Data Mining:

Modellierung, Methodik und Durchführung

ausgewählter Fallstudien

mit dem SAS® Enterprise Miner™

Diplomarbeit

für die Prüfung für Diplom-Volkswirte

eingereicht beim

Prüfungsausschuss für Diplom-Volkswirte

der

Fakultät für Wirtschafts- und Sozialwissenschaften

der

Universität Heidelberg

2003

Christian Gottermeier

geboren in Heidelberg

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe, und dass alle wörtlich oder sinngemäß aus Veröffentlichungen entnommenen Stellen dieser Arbeit unter Quellenangabe einzeln kenntlich gemacht sind.

Christian Gottermeier

I N H A L T S V E R Z E I C H N I S

1. Einführung 1

2. Data Mining 3

2.1 Definitionen und Erklärungen 3

2.2 Einführung in die wichtigsten Verfahren 5

2.2.1 Data Mining als interdisziplinäre Wissenschaft 5

2.2.1.1 Multivariate Analysemethoden 6

2.2.1.1.1 Regressionsanalyse 6

2.2.1.1.2 Clusteranalyse 6

2.2.1.2 Künstliche Intelligenz (KI) und maschinelles Lernen 6

2.2.1.2.1 Entscheidungsbaumverfahren 7

2.2.1.2.2 Künstliche neuronale Netze (KNN) 7

2.2.1.2.3 Selbstorganisierende Karten (SOM) / Kohonen-Netze 7

2.2.1.3 Assoziations- und Sequenzanalyse 8

2.2.2 Alternative Einordnungsmöglichkeiten 8

2.2.2.1 Überwachtes vs. unüberwachtes Lernen 8

2.2.2.2 Parametrische vs. nichtparametrische Verfahren 9

2.3 Architekturüberlegungen 10

2.3.1 Data Warehouse (DWH) und Data Marts 10

2.3.2 Integration mit Data Mining 11

2.3.3 OLAP 11

2.3.4 OLAP und Data Mining 12

2.4 Einsatzgebiete 13

2.4.1 Customer Relationship Management (CRM) 14

2.4.2 Text Mining 14

2.4.2 Web Mining 15

3. Pre-Processing 17

3.1 Partitionierung der Daten 17

3.1.1 Trainings-, Validierungs- und Testdaten 17

3.1.2 Seltene Zielereignisse 17

3.1.3 Massiv große oder beschränkt kleine Datensätze 18

3.1.3.1 Cross Validation 18

3.1.3.2 Sampling 18
3.2 Variablenselektion oder das Problem hoher Dimensionalität 19
3.3 Fehlende Werte 19
3.4 Transformationsprozesse 20

4. Die Methoden 21
Vorbemerkungen: Grundproblematik Generalisierbarkeit 21
4.1 Regressionsanalyse 22
4.1.1 Einführung in die lineare Regression 22
4.1.1.1 Lineare Einfachregression 22
4.1.1.1.1 Schätzung der Koeffizienten 22
4.1.1.2 Lineare Mehrfachregression 23
4.1.1.3 Annahmen des linearen Regressionsmodells 24
4.1.2 Logistische Regression 24
4.1.2.1 Einführung in die logistische Regression 24
4.1.2.2 Der Rechenansatz der logistischen Regression 25
4.1.2.3 Schätzung der Koeffizienten 25
4.1.3 Variablenauswahlverfahren 26
4.2 Clusteranalyse 26
4.2.1 Einführung in die Clusteranalyse 26
4.2.2 K-Means-Verfahren 27
4.2.3 Der K-Means-Algorithmus 27
4.3 Entscheidungsbaumverfahren 28
4.3.1 Aufbau eines Entscheidungsbaums 28
4.3.1.1 Algorithmen 29
4.3.1.2 Auswahlmaße 29
4.3.1.2.1 Informationsgewinn und Entropie 30
4.3.1.2.2 Gini-Index 30
4.3.1.2.3 χ^2-Maß 31
4.3.1.3 Stoppkriterien 32
4.3.2 Pruning 32
4.3.3 Surrogat-Splits für das Einfügen fehlender Werte 33
4.3.4 Wälder: Bagging und Boosting 33
4.4 Künstliche neuronale Netze 35
4.4.1 Einführung in die künstlichen neuronalen Netze 35

4.4.2 Netzwerkarchitektur 35
4.4.2.1 Multilayer Perceptron (MLP) 35
4.4.2.2 Radiale-Basisfunktionen-Netze (RBF-Netze) 39
4.4.3 Lernregel 42
4.4.3.1 Gradientenabstiegsverfahren 43
4.4.3.1.1 Probleme bei Gradientenverfahren 43
4.4.3.2 Backpropagation 44
4.4.3.3 Konjugierter Gradientenabstieg 46
4.4.3.4 Newton-Verfahren 46
4.4.3.5 Levenberg-Marquard 46
4.4.4 Regulierbarkeit 46
4.4.4.1 Early Stopping 46
4.4.4.2 Weight Decay 47
4.4.5 Selbstorganisierende Karten (SOM) / Kohonen-Netze 47
4.4.5.1 Prinzipien der selbstorganisierenden Karten 47
4.4.5.2 Lernverfahren der selbstorganisierenden Karten 48
4.5 Assoziations- und Sequenzanalyse 49
4.5.1 Einführung in die Assoziationsregeln 49
4.5.1.1 Support 50
4.5.1.2 Konfidenz 50
4.5.1.3 Lift 50
4.5.2 Sequenzmuster 51

5. Modellbewertung 52
5.1 Bewertung der Klassifizierungsleistung 52
5.2 Draw Lift Charts 53

6. Fallstudien 55
6.1 Fallstudie A: Optimierung einer Mailing-Aktion 55
6.2 Fallstudie B: Funktionsweise der KNN 58
6.2.1 Auswahl der Netzwerkarchitektur bei NRBF-Netzen 58
6.2.2 Auswahl des Lernverfahrens bei MLP-Netzwerkarchitekturen 60
6.2.3 Early Stopping 63
6.3 Fallstudie C: Entscheidungsbaumverfahren 64
6.3.1 Bestimmung des Auswahlmaßes 64
6.3.2 Bagging 67

7. Zusammenfassung.....68

Anhang.....V

Abbildungsverzeichnis.....XLIX

Literaturverzeichnis.....LII

Abkürzungsverzeichnis.....LIV

1. Einführung

Entscheidungen sind ein Akt des menschlichen Verhaltens, bei denen eine Festlegung für eine unter mehreren Möglichkeiten stattfindet. Da bei diesen Handlungen die Berufung auf Traditionen oder Autoritäten oftmals nicht möglich ist, wurde schon früh auf verschiedenste Hilfsmittel zurückgegriffen. So ließ sich Julius Cäsar von einem Würfelergebnis leiten, General Wallenstein von einem Astrologen beraten oder es wurden Prognosen mit Hilfe von Glaskugeln, Spielkarten oder dem Stand der Sterne getroffen.

Unter wirtschaftlichen Gesichtspunkten sind Entscheidungen eine rationale Wahl zwischen mehreren Möglichkeiten, wobei der Entscheidungsprozess als tragendes Element der ökonomischen Tätigkeit herausgestellt wird. Gerade in diesem Umfeld wird die Entscheidungsfindung – nun allerdings wissenschaftlich fundiert und mit weitreichenden Konsequenzen – durch folgende Verfahren unterstützt: Analysemethoden wie Benchmarking, Lebenszyklus- oder Erfahrungskurvenkonzept und Prognoseverfahren wie die Delphi-Methode oder die Szenario-Technik. Allerdings sind die meisten dieser Verfahren i.d.R. auf spezielle Problemstellungen ausgerichtet. Ganzheitliche Lösungsansätze werden seit den 60er Jahren zur Unterstützung des Managements bereitgestellt. Mit Hilfe von Informationssystemen soll die Entscheidungsfindung verbessert werden. Häufig wechselnde Schlagworte wie z.B. Management Information System (MIS) oder Decision Support System (DSS) konnten allerdings noch keine durchschlagenden Erfolge erzielen. Seit Mitte der 90er Jahre wurden mit neuen konzeptionellen Ansätzen, die meist unter dem Oberbegriff „Business Intelligence“ zusammengefasst werden, erfolgsversprechende Lösungen zum Aufbau entscheidungsorientierter Informationssysteme (EIS) etabliert. EIS setzen sich dabei aus Werkzeugen zur Selektion und Speicherung entscheidungsrelevanter Informationen (Data Warehouse) sowie zur entscheidungsunterstützenden Modellierung (OLAP-Tools) zusammen. Eine konsequente Umsetzung des Data Warehouse Gedanken führt zu immensen Datensammlungen, die, um die Archivierung nicht zum Selbstzweck werden zu lassen, dann auch ausgewertet werden sollen. An dieser Stelle setzt Data Mining an.

In Kapitel 2 werden die Grundzüge des Data Mining dargestellt, eine Verbindung zu Data Warehouse und OLAP gezogen und die Einsatzgebiete skizziert, in denen sich Data Mining durchgesetzt hat. In Kapitel 3 wird der erste wichtige Schritt, der vor der eigentlichen Modellierung stattfinden sollte, das Pre-Processing, erläutert. Die Modelle und die damit verbundenen Methodiken der Data Mining-Verfahren werden in Kapitel 4 vorgestellt. Stets wird eine Verbindung zum SAS® Enterprise Miner™ gesucht und so eine

Anpassung der dort verankerten Möglichkeiten an die Theorie vorgenommen. Die Vorgehensweise der Modellbewertung und die dafür existierenden Kriterien werden in Kapitel 5 dargestellt. Die praktische Umsetzung der Data Mining-Modelle wird anhand verschiedener Fallstudien im sechsten Kapitel gezeigt. Dafür werden die von der SAS® Institute Inc. erstellten Fälle bearbeitet. Diese Daten sind stark idealisiert, d.h. sofort analysierbar und deshalb sehr gut geeignet, um die einzelnen Schritte Pre-Processing, Modellierung der einzelnen Verfahren und Modellbewertung durchzuführen.

2. Data Mining

2.1 Definitionen und Erklärungen

Durch die Entwicklung der Informationstechnologie ist die Datenerhebung, -speicherung und -verwaltung stark ausgeprägt. Datenbanken der Größenordnung Gigabyte oder Terabyte sind weit verbreitet. Schätzungen zufolge verdoppeln sich die weltweit vorhandenen Informationen alle 20 Monate, in Datenbanken ist die Rate noch größer. Allerdings konnten die Möglichkeiten der manuellen Analysetechnik mit dieser Entwicklung nicht Schritt halten. Die Folge ist, dass die Datenmengen nicht zu aussagefähigen Informationen verdichtet werden. Der verschärfte Wettbewerbsdruck zwingt aber zur Nutzung aller Informationsquellen.

Der Begriff Data Mining bezeichnet eine relativ neue Forschungs- und Anwendungsrichtung, obwohl die Bestandteile schon lange existieren. Verfahren der klassischen statistischen Datenanalyse, Anwendungen aus der künstlichen Intelligenz (KI), der Mustererkennung und des maschinellen Lernens wurden in das sog. Knowledge Discovery in Databases (KDD) integriert. Wie der Name schon assoziiert, besteht auch eine starke Verbindung zur Datenbanktechnologie. KDD und Data Mining werden in dieser Arbeit und auch meistens in der Literatur simultan verwendet.

Die Aufgabe des Data Minings ist die Entwicklung, Implementierung und Ausführung von Datenanalysemethoden. Ein Fokus wird dabei auf sehr große Datensätze mit komplexen Strukturen gelegt. Die vier Hauptaufgaben sind:

a.) Vorhersage- und Klassifikationsmodelle,

b.) Segmentierungen oder Clusterung,

c.) Dimensionsreduktion und

d.) Assoziationsanalysen.

Vorhersage und Klassifikation unterscheiden sich hauptsächlich in den Skalierungsmaßen. Ist die abhängige Variable intervallskaliert, spricht man von Vorhersagemodellen. Klassifikationsmodelle dagegen besitzen einen binären, ordinalen oder nominalen Regressand. Man spricht von Klassifikation, weil die Klassenwahrscheinlichkeiten angegeben werden sollen. Vertreter dieser Modelle sind u.a. die Regressionsanalyse, das Entscheidungsbaumverfahren oder die künstlichen neuronalen Netze.

Die Besonderheit des Clusterung ist das Fehlen einer abhängigen Variablen. Als Modelle, die eine Klassenunterscheidung aufgrund der Homogenität bzw. Heterogenität der unabhängigen Variablen vornehmen, können beispielsweise das K-Means-Verfahren oder sog. selbstorganisierende Karten, die Kohonen-Netze, angeführt werden.

Die Dimensionsreduktion ist in komplexen Systemen unerlässlich, um eine Visualisierung der Problemstellung zu ermöglichen. Auch wenn eine Reduktion auf zwei bis drei Dimensionen, die eine grafische Anschauung ermöglicht, nicht gelingt, ist eine Variablenselektion das Instrument, um Modelle zu vereinfachen und damit die Fähigkeit zur Modellentwicklung zu erhöhen (vgl. auch Abschnitt 3.2).

Eine Assoziationsanalyse soll Beziehungen zwischen Variablen aufdecken. Als Verfahren werden Assoziationsregeln, Sequenzmuster oder Link-Analysen verwendet.

In den Vorhersage- und Klassifikationsmodellen liegt der Schwerpunkt dieser Arbeit, sowohl bei der Analyse der Modellierung als auch bei den Fallstudien, bei denen ausschließlich diese Modelle untersucht werden[1].

An dieser Stelle sollte auch auf die Schwierigkeiten eingegangen werden, eine allgemeingültige Definition oder eine einheitliche Terminologie zu finden. Entgegen der Gleichstellung der Begriffe KDD und Data Mining (siehe oben) existiert auch die Sichtweise, KDD als den Gesamtprozess der Analyse anzusehen und Data Mining als Synonym für die einzelnen eingesetzten Methoden zu bezeichnen. Fayyad[2], der das Thema Data Mining seit den Anfängen zu Beginn der 90er Jahre prägte, stellte folgenden Zusammenhang zwischen beiden Begriffen her:

"KDD is the non-trivial process of identifying valid, novel, potential useful and ultimately understandable pattern in data."

"Data Mining is a simple step in the KDD process that under acceptable computational efficience limitations enumerates structures (patterns or models) over the data."

Data Mining mit seinen Methoden aus den verschiedensten Gebieten ist eine interdisziplinäre Wissenschaft. Schon deshalb wird es je nach Schwerpunkt und wissenschaftlicher Herkunft verschiedene Sichtweisen, Ansätze und vor allem Notationen geben. Darüber hinaus finden sich nicht nur in der Literatur, sondern auch in der praktischen Umsetzung der verschiedenen Anbieter von Data Mining-Software unterschiedliche Ansätze, die Teilgebiete ausdrücklich ausgrenzen oder neue Methoden einführen. Der Ansatz dieser Arbeit besteht nicht darin, alle Methoden und Sichtweisen darzustellen, sondern eine Einführung in die Problematik zu bieten, aufgrund derer die nachher folgenden Fallstudien durchgeführt werden.

Die deutsche Data Mining-Übersetzung Datenmustererkennung nennt die Identifizierung von Mustern als Ziel des Data Mining-Prozesses, während der Begriff KDD eine

[1] Der Ablauf bei der Erstellung von Vorhersage- und Klassifikationsmodellen kann im Anhang in Kapitel A.1 nachvollzogen werden.

[2] Küppers (1999), S. 23.

Entdeckung von Wissen impliziert. Häufig wird in Definitionen von Data Mining auch von Informationen gesprochen. In der Literatur sind verschiedene Sichtweisen zur Abgrenzung der Begriffe Daten, Information und Wissen zu finden. So lassen sich die Begriffe beispielsweise auf der Grundlage der Semiotik voneinander abgrenzen. Daten bestehen aus einem oder mehreren Zeichen, die nach vorgegebenen Regeln, der Syntax, aneinander gefügt werden. Daten werden durch einen sinngebenden Zusammenhang, der Semantik, zu Informationen. Wissen entsteht durch die Vernetzung von Informationen mit dem Kontext, der Pragmatik. Es muss also zu einer Bewertung seitens des Anwenders kommen, da das System lediglich verdichtete Daten zur Verfügung stellt. Diese Sichtweise lässt sich in den KDD-Ansatz integrieren. In Kapitel 5 wird die Modellbewertung und damit der Prozess von (verdichteten) Daten zu Informationen dargestellt, die bei einer geeigneten Umsetzung in Wissen transformiert werden kann.

2.2 Einführung in die wichtigsten Verfahren

Bei der Erläuterung der Methoden wird durch die Orientierung am SAS® Enterprise Miner™ ein Praxisbezug[3] hergestellt. So werden künstliche neuronale Netze, die Regressionsanalyse und Entscheidungsbäume dargestellt, außerdem Instrumente für die Clusteranalyse und für Assoziationen. In den folgenden Abschnitten werden die einzelnen Modelle kurz deskriptiv vorgestellt, eine ausführliche Analyse erfolgt in Kapitel 4. In der Literatur finden sich verschiedene strukturelle Ansätze, um die Vielzahl der einzelnen Methoden einzuordnen. In Abschnitt 2.1 wurde die Zusammensetzung der KDD-Analyse aus verschiedenen Forschungsrichtungen erwähnt. Dieser Ansatz wird in dem nächsten Abschnitt aufgegriffen, d.h. die einzelnen Methoden werden den verschiedenen Gebieten zugeordnet und anschließend kurz vorgestellt. In Abschnitt 2.2.2 werden zwei alternative Einordnungsmöglichkeiten skizziert.

2.2.1 Data Mining als interdisziplinäre Wissenschaft

Künstliche neuronale Netze werden der künstlichen Intelligenz (KI) zugeschrieben, das Entscheidungsbaumverfahren als Element des maschinellen Lernens angesehen, was wiederum ein Downgrade der KI ist. Die Regressionsanalyse wird dem Bereich der multivariaten Analysemethoden oder allgemeiner der Statistik zugerechnet, ebenso die Clusteranalyse. Ein eher heuristischer Ansatz ist die Assoziationsanalyse.

[3] Die verschiedenen Knoten des SAS® Enterprise Miner™ werden im Anhang ab Kapitel A.16 beleuchtet.

2.2.1.1 Multivariate Analysemethoden

Multivariate Analysemethoden lassen sich in verschiedene Verfahren[4] einteilen, die sich alle für Data Mining-Prozesse eignen. Welches Verfahren verwendet wird, ergibt sich aus den Skalenniveaus der Variablen. So ist für eine binäre Zielvariable die logistische Regression und im Fall eines intervallskalierten Regressands die lineare Regression vorgesehen. Die Clusteranalyse ist ein Verfahren, das für Segmentierungen vorgesehen ist.

2.2.1.1.1 Regressionsanalyse

Die Regressionsanalyse hat zur Aufgabe, den Zusammenhang zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen aufzudecken. Mit Hilfe der Regressionsanalyse können die unterschiedlichen Beziehungen überprüft und qualitativ abgeschätzt werden. Die Regressionsanalyse ist ein außerordentlich flexibles Verfahren, das sich sowohl für die Erklärung von Zusammenhängen wie auch für die Durchführung von Prognosen bzw. Klassifikationen eignet.

2.2.1.1.2 Clusteranalyse

Die Clusteranalyse ist ein Klassifikationsverfahren. Die Bildung von Klassen erfolgt auf der Grundlage der Einteilung von Merkmalsträgern in Teilmengen. Die Teilmengen sind nicht a priori gegeben, sondern müssen erst gebildet werden. Die Clusterbildung erfolgt nach der Vorgabe, dass die Objekte eines Clusters sich möglichst ähnlich sind, während sich Objekte verschiedener Cluster möglichst deutlich unterscheiden sollen.

2.2.1.2 Künstliche Intelligenz (KI) und maschinelles Lernen

Die künstliche Intelligenz ist ein Teilgebiet der Informatik, welche versucht, menschliche Vorgehensweisen der Problemlösung auf Computern nachzubilden, um auf diesem Weg neue oder effizientere Aufgabenlösungen zu erreichen. Für das Verständnis komplexer Systeme werden häufig Modelle gebildet, in denen die Wirkungszusammenhänge vereinfacht so präzise wie möglich wiedergegeben werden. Dieses auch sog. induktive Lernen ermöglicht die Darstellung menschlicher Denkstrukturen. Die Forschungsrichtung des maschinellen Lernens verfolgt die Zielrichtung einer Automation der Lernprozesse, bei der für einen Datenbestand eine vereinfachte Beschreibung zu finden ist. Im Rahmen dieser Arbeit werden aus dem Bereichen KI und maschinellem Lernen die Verfahren der Entscheidungsbäume, der künstlichen neuronalen Netze, sowie die Spezifikation der

[4] Regressions-, Varianz-, Cluster-, Diskriminanz- und Faktorenanalyse.

selbstorganisierenden Karten vorgestellt. An dieser Stelle sei noch auf das fallbasierte Schließen (Case-Based Reasoning) hingewiesen, dass im SAS® Enterprise Miner™ implementiert wurde.

2.2.1.2.1 Entscheidungsbaumverfahren

Entscheidungsbäume ermöglichen eine weitgehend automatisierte Objektklassifizierung, außerdem können sie zu Vorhersagen genutzt werden. Im Sinne des maschinellen Lernens sind Entscheidungsbäume eine spezielle Darstellungsform von Entscheidungsregeln. Diese bilden strukturelle Zusammenhänge in Daten ab, sind verständlich und leicht nachvollziehbar.

Entscheidungsbäume sind nach dem Top-Down-Prinzip generiert. Beginnend bei dem sog. Wurzelknoten wird in jedem Knoten solange ein Attribut abgefragt und eine Entscheidung getroffen, bis ein Blatt erreicht wird. Bei einem Blatt handelt es sich um einen Knoten, an dem keine weitere Verzweigung durchgeführt wird, hier kann die Klassifikation abgelesen werden. In jedem Schritt wird genau das Attribut gesucht, welches allein die Klassifikation auf den betrachteten Daten am besten erklärt. Dieses Attribut wird dann zur Aufteilung der Daten in Untermengen verwendet, welche anschließend separat betrachtet werden.

2.2.1.2.2 Künstliche neuronale Netze (KNN)

Künstliche neuronale Netze (KNN), Artificial Neural Networks oder schlicht neuronale Netze sind informationsverarbeitende Systeme, die aus einer großen Anzahl einfacher Einheiten (Zellen, Neuronen) bestehen, welche sich Informationen in Form der Aktivierung der Zellen über gerichtete Verbindungen zusenden. Das wesentliche Element der KNN ist ihre Lernfähigkeit, also das selbstständige Lernen aus Trainingsbeispielen.

2.2.1.2.3 Selbstorganisierende Karten (SOM) / Kohonen-Netze

Die selbstorganisierenden Karten, Self-Organizing Maps (SOM), nach ihrem Entwickler Teuvo Kohonen auch Kohonen-Netze oder Kohonen Feature Maps genannt, sind eine Form neuronaler Netze, die in der Lage sind, ein klassifizierendes Verhalten ohne vorgegebene Trainingsausgaben zu erlernen. SOM nutzen dafür die räumliche Anordnung und die Nachbarschaftsbeziehung der Neuronen aus.

2.2.1.3 Assoziations- und Sequenzanalyse

Assoziationsregeln beschreiben Muster von zusammenhängend auftretenden Elementen innerhalb eines Datensatzes. Dieses Verfahren basiert also auf der Häufigkeitsbetrachtung von Attributkombinationen. Entscheidend ist nun, wie prozentual häufig die Attribute gemeinsam innerhalb des gesamten Datenbestandes vorkommen und ob eine ausreichend hohe Konfidenz dafür gefunden werden kann. Ansonsten ergeben sich fast beliebig viele Assoziationen.

Sequenzmuster erweitern diese Regeln um die Dimension Zeit, d.h. sie beschreiben Assoziationsregeln in zusammengehörigen Datensätzen im Zeitverlauf.

Assoziationsregeln dienen mit anderen Worten dem Aufspüren von intratransaktionellen Mustern, während Sequenzmuster intertransaktionelle Muster beschreiben.

2.2.2 Alternative Einordnungsmöglichkeiten

2.2.2.1 Überwachtes vs. unüberwachtes Lernen

Das wichtigste Element beim überwachten Lernen[5] ist der Klassifikator, der verschiedene Objekte in vorgegebene Kategorien[6] einordnet. Mit einer Basis von bekannten Fällen, deren betrachtete Objekte bereits klassifiziert sind, wird das Lernen praktiziert. Das ausgewählte, aber noch nicht angepasste Modell muss nun so konfiguriert werden, dass die bekannten Fälle möglichst gut reproduziert werden können. In diesem Zusammenhang wird meist auch von Generalisierung gesprochen. Dies geschieht durch das Training des Modells. Der Trainingsprozess ist iterativ, d.h. durch verschiedene Lernschritte kommt es zu einer ständigen Verbesserung des Systems. Nach Abschluss des Trainingsprozesses ist es möglich, das Modell auch auf nicht klassifizierte Daten anzuwenden. Dieser Vorgang wird auch als Scoring bezeichnet.

Beim unüberwachten Lernen gibt es weder eine Klassifikation noch eine Basis von bekannten Fällen. Die Entdeckung von interessanten Strukturen steht im Mittelpunkt. Hierbei gibt es zwei Möglichkeiten: Segmentierung und Assoziationen. Bei der Segmentierung kommt es zu einer Partitionierung der Daten in Cluster ohne Vorgabe von Klasseneinteilungen. Im Bereich der Assoziationen werden ausschließlich Objekte betrachtet, die einen grundsätzlich vergleichbaren Informationsumfang haben.

---

[5] Vgl. Krahl (1998), S. 61 ff.

[6] Mögliche Kategorien sind binär (z.B.: 0, 1), mehrfach (z.B.: Aaa, Aa, …, Caa oder 1, …, 10), Funktionskonstruktionen wie Wahrscheinlichkeiten und Prognosen mit zeitlicher Komponente.

2.2.2.2 Parametrische vs. nichtparametrische Verfahren

Im Kontext herkömmlicher statistischer Verfahren wird zwischen parametrischen und nicht-parametrischen Verfahren unterschieden[7]. Beide Verfahren dienen zur Modellierung eines (ökonomischen) Zusammenhangs, fordern jedoch unterschiedlich starke Annahmen. So unterstellt man bei den parametrischen Verfahren dem zu modellierenden Zusammenhang eine bestimmte funktionale Form, bei der lediglich noch die Parameter der unterstellten Funktion zu bestimmen sind. Bei den nichtparametrischen Verfahren wird der Zusammenhang modelliert, ohne Annahmen über dessen funktionale Form treffen zu müssen. Die Entscheidung, welches der Verfahren sich für die Problemstellung eignet, bestimmt sich aus der Kenntnis der zugrunde liegenden Struktur des Zusammenhangs. Existieren zumindest hinreichend belegte Vermutungen, werden parametrische Verfahren zum Einsatz kommen. So lässt sich beispielsweise die Regressionsanalyse aufgrund der Annahmen, die ein Schätzer besitzen soll, wenn er ein BLUE (Best Linear Unbiased Estimators, vgl. Abschnitt 4.1.1.3) ist, den parametrischen Verfahren zuordnen. Bei den KNN (ab Abschnitt 4.4) ist die Einteilung in das jeweilige Verfahren abhängig von der Netzwerkkomplexität. So entspricht ein neuronales Netz ohne Hidden Layer, das also nur aus Input und Output Layer besteht, einem einfachen linearen Regressionsmodell und ist damit ein rein parametrisches Verfahren. Zwar werden keine expliziten Annahmen über die funktionale Form des zu modellierenden Zusammenhangs getroffen, dafür aber die implizite Annahme, dass der zu modellierende Zusammenhang durch das verwendete Netzwerk approximiert werden kann. Soll auch diese Annahme fallengelassen werden, muss ein Netzwerk gewählt werden, dass jede beliebige Funktion approximieren kann, im Extremfall ein Netzwerk, dass unendlich viele Hidden Units besitzt. Ein hinreichend dimensioniertes Netzwerk-Modell entspricht dann den nichtparametrischen Verfahren. Diese theoretische Sicht lässt sich durch die Werte Bias und Varianz eines Schätzers noch verdeutlichen. Der Bias[8] eines Schätzers ist die Abweichung des Erwartungswertes des Schätzers vom tatsächlichen Wert. Durch den Bias kann also die Entfernung des Funktionsverlaufs der geschätzten Funktion von dem der tatsächlichen Funktion angegeben werden. Die Varianz[9] des Schätzers ist als die durchschnittliche quadrierte Abweichung des Schätzers von seinem Erwartungswert definiert[10].

[7] Vgl. Anders (1995), S. 4 ff.
[8] Vgl. Anders (1995), S. 6.
[9] Vgl. Anders (1995), S. 6.
[10] Eine Herleitung der Varianz und des Bias eines Schätzers findet sich im Anhang in Kapitel A.2.

2.3 Architekturüberlegungen

In der Literatur werden die Themenkomplexe Data Mining, Data Warehouse und OLAP (Online Analytical Processing) häufig gemeinsam dargestellt[11]. Dies erscheint einerseits sinnvoll, da ein implementiertes Data Warehouse die Data Mining-Analyse effizienter gestalten kann. Andererseits ist ein Warehouse für einen KDD-Prozess nicht zwingend notwendig. Eine relationale Datenbank ist ausreichend. OLAP ist eine hervorragende Ergänzung zu Data Mining, da die Analyse mit OLAP-Werkzeugen stark durch den Benutzer gesteuert wird, während gegensätzlich dazu Data Mining-Techniken recht selbstständig funktionieren[12].

2.3.1 Data Warehouse (DWH) und Data Marts

Das Data Warehouse stellt ein unternehmensweites Konzept zur effizienten Bereitstellung und Verarbeitung entscheidungsorientierter Daten dar. Diese Daten weisen im Gegensatz zu den transaktionsorientierten Daten einen hohen Aggregationsgrad auf, haben einen Zeitraumbezug und sind den Bedürfnissen der Entscheidungsträger zur Durchführung ihrer Aufgaben angepasst. Inmon[13] fasst die Forderungen, die ein DWH erfüllen soll, folgendermaßen zusammen: *„A data warehouse is a subject oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions."*

Das DWH kann somit im Wesentlichen durch die Merkmale Themenorientierung, Vereinheitlichung, Dauerhaftigkeit und Zeitorientierung beschrieben werden[14].

Die Informationseinheiten im DWH sind auf die inhaltlichen Kernbereiche einer Organisation („*subject oriented*") fokussiert. Die Daten in einem DWH werden zu einem einheitlichen und konsistenten („*integrated*") Datenbestand zusammengefasst. Diese Vereinheitlichung bezieht sich häufig auf Namensgebung, Skalierung, Kodierung und Variablenbelegung mit dem Ziel, auch bei großer Heterogenität einen konsistenten Datenbestand zu erreichen. Operative Daten finden keinen Eingang in ein DWH. Der Import aktueller Daten wird durch operative Vorsysteme gewährleistet, in denen auch die notwendige Aktualisierung vorgenommen wird. Dagegen wird der in das DWH übernommene Datenbestand nach einer fehlerfreien Übernahme nicht mehr verändert, die Daten sind dann dauerhaft („*nonvolatile*") vorhanden. Die Speicherung der Daten erfolgt in den operativen Vorsystemen innerhalb kürzester Zeiträume, im DWH dagegen über

[11] Beispielsweise in Lusti (2002): Data Warehouse und Data Mining.
[12] Eine mögliche Kombination der verschiedenen Ansätze wird im Anhang in Kapitel A.3 gezeigt.
[13] Inmon (1996), S. 3.
[14] Vgl. SAS Institute Inc. (2001), S. 1-17.

längere Zeiträume („*time variant*“) hinweg. Dadurch sind im DWH immer Zeitelemente vorhanden. Hierzu muss die zeitpunktgenaue Betrachtung aus den operativen Systemen mit der lediglich zeitpunktbezogenen Korrektheit in eine zeitraumbezogene Betrachtungsweise transformiert werden.
Eine Variante innerhalb des DWH-Konzepts stellen die Data Marts dar, die als eine kleinere Version eines Warehouses gelten. Data Marts halten lediglich bestimmte Informationen bereit, häufig nur einen Teilausschnitt.

2.3.2 Integration mit Data Mining

KDD und DWH können im Zuge einer geeigneten Strategie zur Verbesserung bei Entscheidungsfindungen in verschiedenen Geschäftsprozessen genutzt werden. Data Mining verlangt nicht zwingend nach einer DWH-Architektur, sie ist aber überaus hilfreich. Es werden nicht nur die benötigten Daten zur Verfügung gestellt, sondern bereits in integrierter und konsistenter Form, die in dieser Qualität sonst nur mit großem Aufwand aus den operativen Systemen direkt beziehbar wären.
Ein Vorteil eines Data Mining mit den Daten eines DWH gegenüber Lösungen ohne ein solches MIS ist, dass Data Mining-Werkzeuge auf Daten zugreifen können, die erst im DWH bereinigt und konsolidiert werden, ohne dass die operativen Datenbanken durch den Data Mining-Prozeß belastet werden[15]. Anschließend können Ergebnisse des KDD-Prozesses direkt zurück auf das DWH übertragen werden.

2.3.3 OLAP

OLAP (Online Analytical Processing)[16] beinhaltet im Wesentlichen die konzeptionelle Basis für Lösungen zur Unterstützung einer dynamischen Datenanalyse. Das Grundprinzip basiert auf der Betrachtung von Daten aus verschiedenen Blickwinkeln (Dimensionen), die eine schnelle und flexible Analyse ermöglichen, welche den Umgang mit großen Datenmengen vereinfacht. Analysen mit OLAP erlauben daher eine multidimensionale Sicht auf die zugrunde liegenden Informationen[17]. Die individuelle und flexible Sichtweise auf die benötigten Daten wird durch die Isolierung einzelner Schichten aus dem gesamten Datenpool ermöglicht. Hierbei wird das Slice & Dice-Konzept genutzt. Slice beschreibt

[15] Vgl. Krahl (1998), S. 51 ff.
[16] Vgl. SAS Institute Inc. (2001), S. 6-32 ff.
[17] Einen OLAP-Würfel, der Umsatzangaben bezüglich Produktlinie, Zeitraum und Region anbietet, wird im Anhang, Kapitel A.4 unter Abschnitt 4.1 gezeigt.

dabei das „Schneiden“ eines bestimmten Ausschnitts aus der Zeitdimension[18]. Neue Perspektiven werden mittels Dice, also durch „Drehen“, „Kippen“ oder „Würfeln“, erreicht. Die Navigation durch die verschiedenen Konsolidierungsebenen[19] erfolgt bequem durch Drill Down, Roll Up oder Drill Across.

2.3.4 OLAP und Data Mining

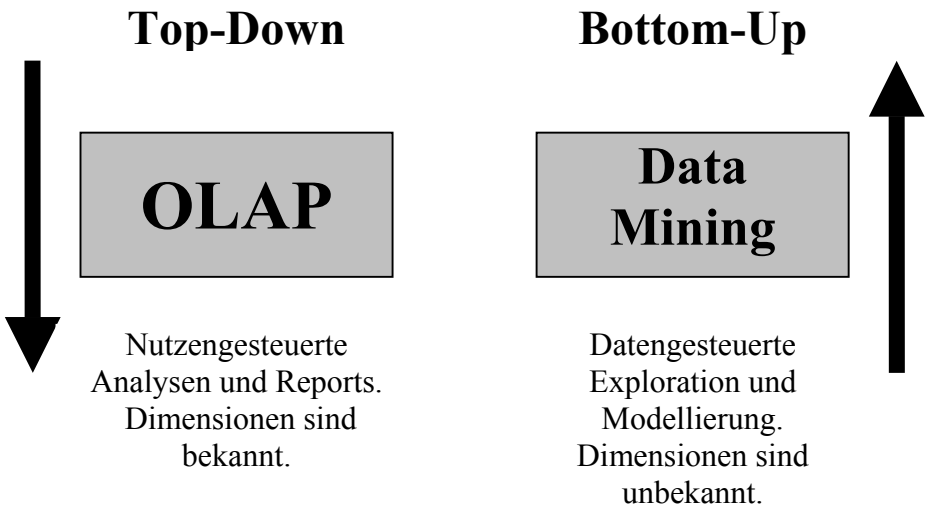


Abb. 2.1: Top-Down- und Bottom-Up-Analyse – OLAP und Data Mining.
Quelle: Vgl. Hofmann, Mertiens (2000), S. 193; eigene Darstellung.

Mit welchem Ansatz die Daten im Warehouse evaluiert werden sollen, hängt stark von der Fragestellung[20] ab. OLAP arbeitet mit bereits definierten Dimensionen und Zusammenhängen. Data Mining dagegen forscht nach neuen, bisher unbekannten Mustern (vgl. Abschnitt 2.1). OLAP und Data Mining sind allerdings keine konkurrierenden Ansätze, sondern ergänzen sich. So können OLAP-Ergebnisse den Wunsch auslösen, die Muster zu verstehen, die hinter den Zusammenhängen stehen. Dieses würde dann eine Data Mining-Analyse nach sich ziehen. Umgekehrt wiederum können neu entdeckte Muster die Basis für OLAP bilden.

[18] Eine Darstellung des Slice-Verfahrens findet sich im Anhang, Kapitel A.4 unter Abschnitt 4.2. Es zeigt jeweils die Perspektive eines Regionalleiters, eines Produktleiters und eines Controllers, außerdem wird eine Ad-Hoc-Sichtweise präsentiert.
[19] Für die Dimension Zeit bedeutet Drill Down eine Abstufung von Quartal, Monat, Woche, etc., bzw. umgekehrt für Roll Up. Drill Across bezeichnet das Blättern durch die verschiedenen Monate.
[20] Vgl. Hofmann, Mertiens (2000), S. 193 ff.

2.4 Einsatzgebiete

Grundsätzlich lassen sich Data Mining-Methoden immer dann sinnvoll anwenden, wenn komplex strukturierte Datenmengen mit erheblicher Größe untersucht werden sollen. Auch wenn in dieser Arbeit ein ökonomischer Fokus gesetzt wird, sollte auf die Möglichkeiten für naturwissenschaftliche Disziplinen[21] verwiesen werden.

Data Mining wird in allen Branchen und betrieblichen Bereichen eingesetzt[22]. Es erweist sich hier als ein wichtiges Werkzeug im operativen Managementprozess[23]. Auch bei der Umsetzung der Unternehmensstrategie liefert KDD wichtige Erkenntnisse. Dies zeigt sich auch daran, dass Data Mining Einzug in die verschiedenen neuen Managementtheorien gefunden hat. Das benötigte Wissen, um Konzepte wie Balanced Scorecard (BSC)[24], Supply Chain Management (SCM)[25], Total Quality Management (TQM) bzw. Six Sigma[26], Supplier Relationship Management (SRM)[27] oder Customer Relationship Management (CRM) umzusetzen, wird durch den Data Mining-Prozess „Pre-Processing, Modellanpassung und Auswertung" entwickelt. Da sich Data Mining bislang im CRM am stärksten durchgesetzt hat, werden die Möglichkeiten, die sich dort bieten, im nächsten Abschnitt kurz vorgestellt.

---

[21] Prozesssteuerung und -kontrolle in der Elektrotechnik, Genomentschlüsselung in der Biologie und Auswertungen im Bereich der Medizin.

[22] Vgl. Küppers (1999), S. 123 ff.

[23] Ein kleiner Überblick ohne Anspruch auf Vollständigkeit wird im Anhang in Kapitel A.5 geboten.

[24] Das Konzept der BSC ist eine konsequente Ausrichtung der Maßnahmen auf ein gemeinsames Ziel. Visionen und Strategien sollen nicht nur durch Finanzkennzahlen überprüft werden. Interne Geschäftsprozesse, Kunden, sowie das Lernen und Entwicklung im Unternehmen werden gleichberechtigt zum finanziellen Umfeld bewertet.
Vgl. Kaplan, Norton (2001), S. 95 ff.

[25] SCM ist die Koordination einer strategischen und langfristigen Zusammenarbeit mit Partnern im gesamten Logistiknetzwerk zur Entwicklung und Herstellung von Produkten, wobei jeder Partner in seinen Kernkompetenzen zuständig ist.
Vgl. Schönleben (2000), S. 53.

[26] TQM ist eine auf der Mitwirkung aller ihrer Mitglieder basierende Managementmethode einer Organisation, die Qualität in den Mittelpunkt stellt und durch Zufriedenheit der Kunden auf langfristigen Geschäftserfolg sowie auf Nutzen der Mitglieder der Organisation zielt.
Vgl. Hummel, Malorny (2002), S. 5 ff.
Six Sigma ist eine moderne Weiterentwicklung, bei der die Qualitätsmessung stärker von analytischen und statistischen Kennzahlen geprägt ist.

[27] SRM wirkt auf den Aufbau und die Optimierung der Lieferantenbeziehung hin.

2.4.1 Customer Relationship Management (CRM)

Die Marketing-Bedingungen haben sich verändert: Gesättigte Märkte und die daraus resultierende veränderte Nachfrage, kritische und gut informierte Kunden, sowie ein hoher Wettbewerbs- und Kostendruck. Diese Faktoren bewirkten einen Paradigmenwechsel[28]. Die ungezielte Streuung von Maßnahmen im Marketing wird zunehmend durch die Strategie abgelöst, den Kunden in den Mittelpunkt der Marketingaktivität zu stellen. Demnach bezeichnet Customer Relationship Management die profitable Kunst, einen Kunden nicht nur zu gewinnen, sondern ihn dauerhaft an die Produkte und Services eines Unternehmens zu binden.[29] Die dafür notwendigen strategischen Entscheidungen sollten nicht als Einzellösungen begriffen, sondern ihre Umsetzung durch Initiativen, die unternehmensweit ergriffen werden, stattfinden. Data Mining-Verfahren korrespondieren, wie in den Abschnitten 2.3.2 und 2.3.4 dargelegt, mit Data Warehouse-Architekturen und OLAP-Analysen. CRM bietet in seiner Vielfalt an Verfahren[30] die größten Anwendungsmöglichkeiten.

2.4.2 Text Mining

Der KDD-Ansatz, aus Datenbanken Muster zu generieren, um damit Informationen zu gewinnen und schließlich Wissen zu erzeugen, ist auf numerische Daten wie Finanzzahlen, Kundendaten oder sonstige Kennzahlen beschränkt. Daraus lassen sich zwar wichtige Rückschlüsse für die strategische Ausrichtung und Steuerung eines Unternehmens ziehen, die Gründe für die bisherige Entwicklung bleiben allerdings häufig im Verborgenen. Ein Großteil der unternehmenskritischen Informationen ist nämlich in Textdokumenten enthalten. Dazu gehören E-Mails, Memos und Presseerklärungen, sowie Strategiepapiere und Beiträge aus Tages- und Fachpresse. Im Zusammenhang mit CRM sind Call Center-Aufzeichnungen und Kundenkorrespondenz von entscheidender Wichtigkeit. Diesbezügliche Informationen konnten bisher nicht analysiert werden, da unstrukturierte Textdaten sich nicht wie numerische Fakten systematisch auswerten ließen. Einen Lösungsansatz bietet Text Mining-Software, die im Text verborgenes Wissen identifiziert,

[28] Durch den Durchbruch des CRM setzt sich die Sichtweise, dass Käufer nicht mit Kunden gleichzusetzen sind, immer mehr durch. So wandelt sich das Ziel von „to make a sale“ zu „to make a customer“. Der Verkauf, bisher Abschluss, wird nun zu dem Beginn der Kundenbeziehung.

[29] Vgl. Hofmann, Mertiens (2000), S. 189.

[30] Klassifikation der Kunden und deren Verhalten, Kunden-Akquisition, Analysen über Kundeninteresse und –profile, Kundenbindung und Churn-Management, Warenkorbanalyse und Aufdeckung von Up- und Cross-Selling-Potenzialen, Optimierung des Call Center- und Kampagnen-Management, Produktentwicklung entsprechend der Kundenanforderungen einschließlich Just-in-time-Fertigung und Versand, Response-Optimierung und die verschiedenen Methoden des Database Marketing.

indem sie die Strukturen des Dokuments analysiert. Auf diese Weise lassen sich Zusammenhänge zwischen einzelnen Dokumenten innerhalb einer größeren Textsammlung entdecken. Gruppierungen lassen sich aufgrund der Häufigkeit einzelner Wörter und Wortkombinationen vornehmen. Texte, die aufgrund enthaltener Wörter als ähnlich erkannt werden, lassen sich zusammenfassen. Ein anderes Verfahren ist das selbstständige Zuordnen von neuen, nicht klassifizierten Dokumenten zu vorab definierten Gruppen. Das Klassifizieren von Texten ist besonders geeignet, um große Mengen an neuen Informationen zu filtern, ohne Schlüsselwörter definieren zu müssen.

Text Mining bezieht sich auch auf Erfahrungen der Linguistik, die besagen, dass Sprache unabhängig vom jeweiligen Dokument eindeutig gegliedert ist, z.B. in der Wort- und Satzbildung.

Die möglichen Einsatzgebiete von Text Mining sind ebenfalls gewaltig, vielleicht sogar noch umfangreicher als die des reinen Data Minings. Eingesetzt wird es beispielsweise schon im Beschwerdemanagement, im Rahmen einer optimalen Kundenansprache inklusive maßgeschneiderter Produktangebote oder in E-Mail-Systemen, die Nachrichten automatisch an den zuständigen Mitarbeiter weiterleiten. Durch den Siegeszug des Internets ist die Bedeutung von Texten als Träger strategisch wertvollen Wissens noch einmal gewachsen: Auskünfte zur Marktsituation oder über Mitbewerber sind nun ohne Aufwand – per Mausklick – erhältlich und lassen sich Profit bringend nutzen.

2.4.3 Web Mining

Data Mining-Verfahren sind grundsätzlich auch für den Einsatz im Bereich des Internets geeignet[31], da dort riesige Datenmengen bereitstehen bzw. sich erheben lassen. Die Grundlage einer Datenanalyse sind durch Logfile-Informationen, Cookies[32] und Einträge im Application Log[33] gegeben. In den operativen Browsing-Informationen des Logfiles finden sich Einträge über die IP-Adresse[34] des Nutzers, HTML-Befehle, die URL der

[31] Vgl. Oesterer (2002), S. 12.

[32] Cookies sind kleine Protokolldateien, die von den Servern im Internet auf die Festplatte des Benutzers abgelegt werden. Anhand der Cookies lassen sich Rückschlüsse bezüglich der Verweildauer auf den verschiedenen Servern ziehen und diese für Benutzerprofile verwenden.

[33] Große Websites, Portale oder Shop-Lösungen beherbergen häufig komplexe Anwendungen meistens unter Anwendung von Applikationsservern. Um das dort stattfindende Benutzerverhalten protokollieren zu können, sind Einträge in das Application Log erforderlich und lassen sich für eine Datenauswertung nutzen.

[34] Die IP-Adresse besteht aus zwei Elementen: der Netzwerkadresse und der Hostadresse. Die Netzwerkadresse gibt die Kennung des Netzwerks an. Die Hostadresse nimmt die Kennung eines bestimmten Geräts im Netzwerk vor.

aufgerufenen Seite, über das Protokoll (meist HTTP) und eine User-ID. Informationen lassen sich außerdem aus Login-Daten[35] und Session-IDs[36] gewinnen.

Der Begriff „Web Mining“ beschreibt demnach den Einsatz von Data Mining-Algorithmen in den oben beschriebenen Datenbereichen. Web Mining lässt sich in Web Content Mining und Web Usage Mining unterteilen[37]. Das Web Content Mining kann im weitesten Sinne als die Übertragung von Text Mining-Methoden auf Web-Seiten im Internet verstanden werden. Das Ziel ist die Gewinnung interessanter Erkenntnisse aus im Internet verfügbaren Dokumenten. Das Interesse des Web Usage Mining richtet sich hingegen primär auf die Analyse des Navigationsverhaltens der Internet-Nutzer, die auch als Clickstream-Analyse bezeichnet wird. Einsatzgebiete, die sich für Web Mining eignen, sind beispielsweise Personalisierung von Webseiten und die Optimierung der Webseitengestaltung bezüglich des Einsatzes von Werbebannern. Die Klassifikation der Kunden nach ihrem Informations- und Einkaufsverhalten ist eine weitere Rendite versprechende Anwendungsmöglichkeit.

[35] Login-Daten werden häufig bei in Anspruchnahme von Diensten erhoben und können Angaben u.a. zur Person, dem Haushalt und dem ökonomischen Umfeld fordern. Allerdings muss der Wahrheitsgehalt bezweifelt werden. Die E-mail-Adresse kann durch die Bestätigung des Passworts das einzig richtige Ergebnis darstellen, hier allerdings mit der Einschränkung, dass diese Adresse evt. nur für die Anmeldung verwendet worden ist.

[36] Eine Session-ID wird jedem Besucher beim erstmaligen Aufruf der Webpage zugeordnet und unterscheidet ihn damit von anderen Besuchern.

[37] Ein weiteres Teilgebiet des Web Minings ist das Web Structure Mining, das sich mit der Organisation und Struktur des World Wide Web beschäftigt. Aufgrund des eher geringen ökonomischen Nutzens wird dieses Gebiet im Folgenden nicht weiter betrachtet.

3. Pre-Processing

Daten, die für die Entwicklung von Vorhersagemodellen verwendet werden, wurden aus operationalen Gründen gesammelt. Diese Daten entsprechen jedoch nicht den Ansprüchen und Forderungen, die für statistische Analysen getroffen werden. In diesem Zusammenhang wird auch häufig von „*opportunistischen*“[38] Daten gesprochen. So sind operational erhobene Datensätze von massiver Größe, dynamisch und häufig verrauscht. Vor der Modellentwicklung muss der zu untersuchende Datensatz ausgewertet und ggf. modifiziert werden, was dem Explore- und Modify-Schritt der SEMMA-Methode[39] entspricht.

3.1 Partitionierung der Daten

3.1.1 Trainings-, Validierungs- und Testdaten

Soll ein Vorhersagemodel neue Daten generalisieren, bedarf es bei den Daten, bei denen bekannt ist, welchen Einfluss die unabhängigen Daten auf den Regressand haben, eine Einteilung in Trainings-, Validierungs- und Testdaten. Unter Training versteht man die Anpassung der Daten, so dass der gewünschte Output erzielt wird. Anschließend wird überprüft, ob die Validierungsdaten tatsächlich auch das Ergebnis der Trainingsdaten bestätigen, ansonsten wird das Modell weiterentwickelt. Die Testdaten werden für eine abschließende Überprüfung zur Verfügung gestellt. Der prozentuale Anteil der Partitionierung[40] ist abhängig von der Datenqualität.

3.1.2 Seltene Zielereignisse

In Vorhersagemodellen sind häufig die interessanten Fälle, also die Ereignisse, die man prognostizieren möchte, selten existent[41]. Beispiele sind die Betrugserkennung, das Credit Scoring oder die Response-Optimierung. Bei diesen Problemstellungen tritt das Zielereignis nur bei weniger als zehn Prozent ein. Normalerweise führt ein Anstieg der Datenmasse zu einer besseren Modellierung, steigen aber lediglich die sog. Non-Event-Fälle, verbessert sich das Modell nicht, sondern führt häufig zu einer Verschlechterung. Deshalb sollte die Datenbasis vor der Analyse modifiziert werden und das Verhältnis Event zu Non-Event im Bereich von 70 zu 30 bis 50 zu 50 liegen. Die Folge dieser

[38] SAS Institute Inc. (2000d), S. 9.
[39] Die SEMMA-Methode wird zusammen mit den einzelnen Konten im Anhang ab Kapitel A.14 vorgestellt.
[40] Bei verrauschten Daten sollte die Hauptgewichtung auf den Trainingsdaten liegen, der verbleibende Anteil für die Validierung genutzt werden. Eine annähernd gleichverteilte Partitionierung kann bei qualitativ guten Datensätzen vorgenommen werden.
[41] Vgl. SAS Institute Inc. (2000d), S. 12 ff.

Operation wird ein Bias sein, der allerdings danach mit einem Offset-Faktor[42] oder Fallgewichten[43] behoben werden kann.

3.1.3 Massiv große oder beschränkt kleine Datensätze

3.1.3.1 Cross Validation

Die Partitionierung der Datenbasis in Trainings-, Validierungs- und Testdaten ist die gängige Methode bei der Modellanpassung. Allerdings kann diese Vorgehensweise bei begrenzten Datensätzen aufgrund mangelnder Repräsentativität zu unzuverlässigen Modellen führen. Ein effizientes Verfahren bei dieser Problematik ist Cross Validation[44]. Der gesamte Datenbestand wird in fünf gleich große Datensätze aufgeteilt. Nun werden vier Datensätze für den Trainingsprozess und der fünfte für die Validierung verwendet. Dieser Prozess wird solange wiederholt bis jeder einzelne Datensatz für die Validierung eingesetzt wurde. Die resultierenden Ergebnisse werden dann gemeinsam für die Modellanpassung benutzt[45].

3.1.3.2 Sampling

Ist die Datenbasis extrem umfangreich, bietet es sich an, Stichproben aus der Grundgesamtheit zu ziehen, da dies eine signifikante Verkürzung der Arbeitsprozesse bewirkt. Beziehungen in einer Stichprobe können generalisiert werden, wenn die Daten ausreichend repräsentativ sind. Darüber hinaus empfiehlt es sich, mehrere Stichproben zu ziehen, um zu verhindern, dass einzelne Merkmale überproportioniert Einfluss auf die Modellanpassung nehmen.

---

[42] Ein Offset-Faktor der Form $\ln\left(\frac{\pi_0\rho_1}{\pi_1\rho_0}\right)$, mit ρ_0 als vorhergesagtem Wert für den Non-Event und ρ_1 als vorhergesagtem Wert für den Event sowie π_0 als tatsächliche Verteilung für den Non-Event und π_1 als tatsächliche Verteilung für den Event ermöglicht die Rückrechnung auf die korrigierten Prognosewerte ρ_1^* und ρ_2^*.

[43] Die vorhergesagten Werte ρ_0 für Non-Event und ρ_1 für Event werden korrigiert, indem die tatsächliche Verteilung π_0 für Non-Event und π_1 für Event durch Fallgewichte der Form:

$$weight_i = \begin{cases} \frac{\pi_1}{\rho_1} \; wenn \; y_i = 1 \\ \frac{\pi_0}{\rho_0} \; wenn \; y_i = 0 \end{cases}$$

dem Ergebnis hinzugefügt werden. Auf diese Weise wird die Wahrscheinlichkeitsverteilung der Event- und Non-Event-Fälle korrekt repräsentiert.

[44] Vgl. SAS Institute Inc. (2000a), S. 80 ff.

[45] Cross Validation wird im Anhang in Kapitel A.6 dargestellt.

3.2 Variablenselektion oder das Problem hoher Dimensionalität

Die Dimension steht in Abhängigkeit zu der Anzahl der Inputvariablen. Die benötigte Rechenleistung für die Modellanpassung wird stärker von der Anzahl der Inputvariablen beeinflusst als von der Anzahl der Fälle. Ein weiteres Problem wird von Breiman als der „Fluch der Dimensionalität“[46] bezeichnet: Eine hohe Dimensionalität begrenzt die Fähigkeit, Beziehungen zwischen Variablen zu entdecken und zu modellieren. Eine höhere Dimensionalität führt zu einem sprunghaften Anstieg der Komplexität der Daten.

Die Lösung dieses Problems ist die Dimensionsreduktion. Irrelevante oder redundante Variablen werden für die Modellentwicklung ignoriert.

Eine Auswahl der Variablen kann durch den R^2- oder den χ^2-Test durchgeführt werden. Weitere Möglichkeiten sind die Hauptkomponentenanalyse[47], das Screening[48] oder das Clustern von Variablen. Außerdem kann das Entscheidungsbaumverfahren dazu verwendet werden, die entscheidungsrelevanten Variablen zu selektieren. An dieser Stelle sei auf Kapitel 4 und die entsprechenden Vorbemerkungen verwiesen, die diese Problematik noch einmal aufgreifen.

3.3 Fehlende Werte

Fehlende Werte besitzen zwei verschieden Problemfelder: Die Behandlungsweise der Missing Values bei dem Aufbau des Modells und schließlich während des Score-Vorgangs. Die einfachste Strategie bei dem Umgang mit fehlenden Werten ist die Complete Case Analysis, bei der nur die Fälle für die Modellanpassung genutzt werden, die vollständig vorhanden sind. Dies kann allerdings bei einer hohen Anzahl an Variablen zu einer hohen Ausschussrate führen, und dies schon bei relativ geringem prozentualem Anteil an Missing Values[49]. Erfolg versprechender ist allerdings eine geeignete Einfügestrategie, die abhängig von den Skalierungsmaßen der Variablen ist. Einfache Methoden wie Mittelwert, Median oder Modus, aber auch komplexere Verfahren der Statistik wie Tukey’s Biweight, Huber oder Andrew’s Wave oder die sog. Cluster Imputation[50] werden im SAS® Enterprise Miner™ für die Einfügung zur Verfügung gestellt. Eine zusätzliche Möglichkeit ist das Bilden von Indikator-Variablen, die ergänzend zur der verwendeten Einfügestrategie angeben, dass an dieser Stelle ein Missing

[46] SAS Institute Inc. (2000d), S. 11.
[47] Die Hauptkomponentenanalyse wird im Anhang in Kapitel A.19, Abschnitt 19.2 erläutert.
[48] Jede Variable wird einzeln auf ihren Wirkungszusammenhang zu der Zielvariable untersucht. Einflüsse von Variablen untereinander werden ebenso wie Interaktionen zwischen Variablen nicht berücksichtigt.
[49] Ein Beispiel der Complete Case Analysis findet sich im Anhang in Kapitel A.7.
[50] Fehlende Werte werden durch Clusterung in Abhängigkeit zu anderen, bekannten Werten gesetzt.

Value vorgelegen hat. Dies kann sinnvoll sein, da die fehlenden Werte systematisch mit den Daten zusammenhängen und so weitere Informationen beinhalten. Einfügestrategien sind auch ein probates Mittel zur Behebung von fehlenden Werten im Score-Vorgang.
Das Entscheidungsbaumverfahren benötigt keine Einfügestrategie, da es die fehlenden Werte, ähnlich den Indikator-Variablen, als eigenständige Ausprägung ansieht. Darüber hinaus kann durch Verwendung von sog. Surrogat-Splits das Einfügen der Missing Values im Score-Vorgang gesteuert werden (vgl. Abschnitt 4.3.3).

3.4 Transformationsprozesse

Ein weiterer wichtiger Schritt, der vor der Modellanpassung bedacht werden sollte, ist die Transformation der Variablen. Folgende Probleme können auftreten: Fehlende Normalverteilung[51], ungünstige Skalierungen oder Probleme mit Ausreißern. Eine Normalverteilung kann durch verschiedene Transformationen[52] erzeugt werden. Skalierungen wie beispielsweise Postleitzahlen haben aufgrund ihrer Intervallskaliertheit nur einen geringen Aussagewert. Dasselbe gilt für viele Variablen, die ordinalskaliert sind. An dieser Stelle müssen Rekodierungen vorgenommen werden. Dies kann durch den Einsatz von Dummy-Variablen oder durch Bildung neuer Variablen geschehen.
Eine weitere Möglichkeit der Verbesserung ist die Behebung der Ausreißer durch Filter. Dies kann ökonomisch signifikant sein. So sorgt beispielsweise ab einem gewissen Niveau ein weiterer Anstieg des Einkommens für keine nennenswerten Steigerungen im Konsum.

[51] Die Normalverteilung wird bei verschiedenen Schätzverfahren vorausgesetzt.
[52] z.B. durch Logarithmieren, Quadrieren, Wurzel ziehen, usw.

4. Die Methoden

Vorbemerkungen: Grundproblematik Generalisierbarkeit

Um eine gute Generalisationsfähigkeit zu erlangen, muss die Modellkomplexität den Daten angepasst werden. Die Problematik verdeutlicht sich bei Betrachtung der Extremfälle:

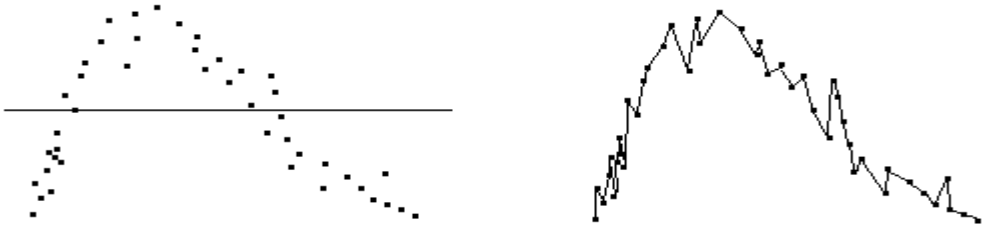


Abb. 4.1: Null-Modell vs. Interpolation.
Quelle: Eigene Darstellung.

Abbildung 4.1 zeigt Modelle, deren Anpassungen an den Datenbestand nicht für das Scoring geeignet sind. Das erste Modell besitzt keine Aussagekraft, es ist ein sog. Null-Modell. Es ist nicht ausreichend komplex, was auch als Underfitting bezeichnet wird. Modelle mit Underfitting zeichnen sich durch einen hohen Bias und eine geringe Varianz aus. Dagegen ist das zweite Modell zu komplex gestaltet worden, so dass eine Interpolation der Daten vorliegt. Dieses Overfitting hat als Merkmale geringen Bias und hohe Varianz. Wie schon in Abschnitt 2.2.2 angedeutet, ist die Bestimmung der Modellkomplexität ein Problem, welches häufig im Zusammenhang mit KNN und der Möglichkeit der Variation bei der Anzahl der Neuronen und den verdeckten Schichten auftritt. Durch die Tatsache, dass der funktionale Zusammenhang i.d.R. vor der Modellanpassung nicht bekannt ist, können die verschiedenen Verfahren stark unterschiedliche Generalisierungsfähigkeiten aufweisen. Entscheidungsbäume mit univariaten Splits können nur senkrechte bzw. waagerechte Trennungen vornehmen. Die lineare Separierbarkeit ist daher mit Bäumen – im Gegensatz zur Regressionsanalyse – nur näherungsweise zu bewerkstelligen. Die Regressionsanalyse dagegen hat trotz Interaktionstermen Schwierigkeiten bei nicht-linearen Abhängigkeiten:

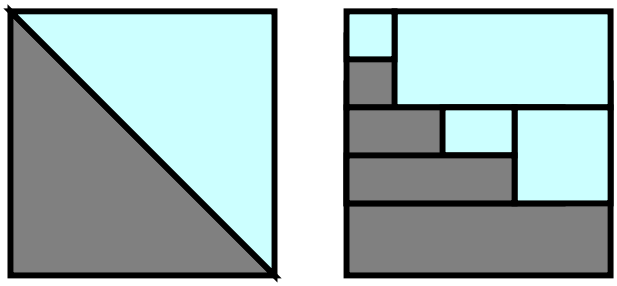


Abb. 4.2: Modellanpassung eines Entscheidungsbaumes.
Quelle: Eigene Darstellung.

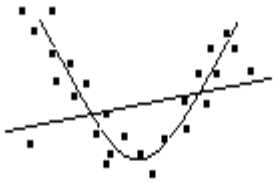


Abb. 4.3: Modellanpassung mit der Regression.
Quelle: Eigene Darstellung.

4.1 Regressionsanalyse

4.1.1 Einführung in die lineare Regression

Mittels der Regressionsanalyse werden die Parameter funktionaler Beziehungen zwischen Variablen geschätzt. Üblicherweise wird eine Kausalrichtung postuliert, so dass der Einfluss der Regressoren auf die erklärende Variable geschätzt wird.

4.1.1.1 Lineare Einfachregression

Es wird angenommen, dass ein linearer Zusammenhang zwischen den beobachtbaren Variablen X und Y besteht. Es wird dabei aber lediglich eine Wirkung von X auf Y unterstellt. Der beobachtete Zusammenhang ist allerdings nicht perfekt, d.h. es gibt noch weitere unbeobachtbare Einflussfaktoren, die keinen systematischen Einfluss auf Y haben und deshalb in der Zufallsvariable ε zusammengefasst werden können. Dadurch wird das Modell stochastisch. Die zugrunde liegende Modellgleichung[53] lautet:

(4.1) $Y_i = \alpha + \beta X_i + \varepsilon_i \qquad (i = 1, \ldots, N).$

Das lineare Regressionsmodell übernimmt damit die Aufgabe einer Identifikationsstrategie zur Lösung kontrafaktischer Fragestellungen.

4.1.1.1.1 Schätzung der Koeffizienten

Die Parameter α und β sind die zu schätzenden Koeffizienten des Modells. Der Parameter β gibt den Einfluss einer marginalen Veränderung der Variable X auf die Variable Y an. Entsprechend der oben verwendeten Argumentation gibt α an, wie groß Y_i wäre, wenn X_i Null wäre. Anschaulich bedeutet die Schätzung von α und β, aus der gegebenen Stichprobe eine Gerade in die Punktwolke, die durch die Beobachtungspaare (Y_i, X_i) entsteht, zu legen, so dass ein festgelegtes Optimierungsziel erfüllt wird. Für die Schätzung der Parameter existieren mehrere Schätzmethoden, wobei die Methode der kleinsten Quadrate (Ordinary Least Squares, OLS)[54] und die Maximum Likelihood-Methode[55] zu den bekanntesten und am weitesten verbreiteten gehören. Das OLS-Prinzip gibt als Optimierungsziel die Minimierung der Summe der quadrierten senkrechten Abstände e der Beobachtungspaare (Y_i, X_i) von der Gerade an. Formal bedeutet dies:

(4.2) $e_i = Y_i - \hat{Y}_i = Y_i - a - bX_i .$

[53] Vgl. Winkler (1997), S. 131.
[54] Vgl. Winkler (1997), S. 132.
[55] Die Maximum Likelihood-Methode wird in Abschnitt 4.1.2.3 besprochen.

Das OLS-Opimierungskalkül ergibt sich dann also formal als:

$$(4.3) \quad \sum_{i=1}^{N} e_i^2 = \left[Y_i - \left(a + bX_i\right)\right]^2 \rightarrow min.$$

Für den Regressionskoeffizienten lautet die Schätzformel:[56]

$$(4.4a) \quad b = \frac{\frac{1}{N}\left(\sum_{i=1}^{N} Y_i X_i\right) - \overline{YX}}{\frac{1}{N}\left(\sum_{i=1}^{N} X_i^2\right) - \overline{XX}} \text{ bzw.}$$

$$(4.4b) \quad b = \frac{Cov\left(X,Y\right)}{s_X^2}.$$

Für das konstante Glied lautet die Schätzformel:

$$(4.5) \quad a = \overline{Y} - b\overline{X}.$$

4.1.1.2 Lineare Mehrfachregression

Analog zur linearen Einfachregression besteht im multivariaten Fall auch ein linearer Zusammenhang, diesmal allerdings zwischen den k beobachtbaren Variablen X_1, X_2, …, X_k und der beobachtbaren Variablen Y.[57] Die Regressionsgleichung[58] für das Individuum i der Stichprobe lautet nun:

$$(4.6) \quad Y_i = \beta_1 + \beta_2 X_{i1} + \beta_3 X_{i2} + \ldots + \beta_k X_{iK} + \varepsilon_i.$$

Daraus folgt das Gleichungssystem:

$$(4.7) \quad \begin{aligned} Y_1 &= \beta_1 + \beta_2 X_{11} + \beta_3 X_{12} + \ldots + \beta_k X_{1K} + \varepsilon_1 \\ Y_2 &= \beta_1 + \beta_2 X_{21} + \beta_3 X_{22} + \ldots + \beta_k X_{2K} + \varepsilon_2 \\ &\vdots \\ Y_N &= \beta_1 + \beta_2 X_{N1} + \beta_3 X_{N2} + \ldots + \beta_k X_{NK} + \varepsilon_N \end{aligned}$$

bzw. in Matrixnotation:

$$(4.8) \quad \boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon}.$$

Die zu schätzenden Parameter des Modells sind also die Elemente des Vektors β. Analog zur linearen Einfachregression kann der OLS-Schätzer für den Parameter β verwendet werden:

$$(4.9) \quad b = \boldsymbol{(X'X)^{-1}X'Y}.$$

---

[56] Mit $\overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$ und $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$.

[57] Ebenfalls wird wieder die Kausalrichtung postuliert und unbeobachtbare Einflussfaktoren, die keinen systematischen Einfluss auf Y haben, in der Zufallsvariable ε zusammengefasst.

[58] Vgl. Stier (1999), S. 243.

4.1.1.3 Annahmen des linearen Regressionsmodells

Das lineare Regressionsmodell $y_k = \beta_0 + \sum_{j=1}^{J} \beta_j x_{jk} + \mu_k$ ist mit einer Reihe von Annahmen[59] verbunden:

- *Linearität:* Das Modell ist linear in den Parametern β_0 sowie β_j, und die Anzahl der Beobachtungen K ist größer als die der Variablen J.
- *Vollständigkeit:* Die Störgrößen haben den Erwartungswert Null, dies impliziert, dass alle erwarteten Variablen im Modell berücksichtigt worden sind ($E(\mu_k) = 0.$).
- *Homoskedastizität:* Die Störgrößen haben die konstante Varianz σ^2 ($Var(\mu_k) = \sigma^2$).
- *Keine Autokorrelation:* Die Störgrößen sind voneinander statistisch unabhängig ($Cov(\mu_k, \mu_{k+r}) = 0$ mit $r \neq 0$).
- *Keine (exakte) Multikollinearität:* Zwischen den erklärenden Variablen besteht keine lineare Abhängigkeit.
- *Die Störgrößen sind normalverteilt.*

Die ersten fünf Annahmen sorgen bei Verwendung der Kleinste-Quadrate-Methode für Schätzwerte, die *BLUE* sind (Best Linear Unbiased Estimators)[60]. Best steht für effiziente, d.h. kleinstmögliche Varianz aufweisende Schätzwerte. Prämissenverletzungen des linearen Regressionsmodells führen im Falle der Nichtlinearität in den Parametern und Unvollständigkeit des Modells zu einer Verzerrung der Schätzwerte. Liegt bei den Störgrößen Autokorrelation oder Heteroskedastizität vor oder besteht Multikollinearität der Regressoren, dann ist die Schätzung ineffizient.

4.1.2 Logistische Regression

4.1.2.1 Einführung in die logistische Regression

Die logistische Regression[61] wird bei binären Zielwerten verwendet, die in jedem Fall durch einen k-dimensionalen Vektor an Input-Variablen bestimmt werden.

Es gilt: $y_i = \begin{cases} 1 & \text{Zielereignis für Fall i.} \\ 0 & \text{kein Zielereignis für Fall i.} \end{cases}$

Die Bestimmung der Ereigniseintrittswahrscheinlichkeiten soll durch folgende funktionale Form erreicht werden:

(4.10) $logit\ (p_i) = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_{ki}$.

[59] Vgl. Backhaus, Erichson, Plinke, Weiber (2000), S. 33.
[60] Vgl. Backhaus, Erichson, Plinke, Weiber (2000), S. 33.
[61] Vgl. SAS Institute Inc. (2000d), S. 16 ff.

Die Ereigniseintrittswahrscheinlichkeit bei gegebenen Inputs ($x_i = x_{1i}, x_{2i}, \ldots, x_{ki}$) lautet:

(4.11) $p_i = E(y_i \mid x_i) = \Pr(y_i = 1 \mid x_i)$.

Das Modell der logistischen Regression nimmt also an, dass der Logit der Ereigniseintrittswahrscheinlichkeit sich aus einer Linearkombination von Inputvariablen zusammensetzt, wobei die unbekannten Parameter β_k geschätzt werden müssen.

4.1.2.2 Der Rechenansatz der logistischen Regression

Eine Linearkombination kann jeden Wert annehmen, während Wahrscheinlichkeiten zwischen Null und Eins liegen. Deswegen wird die Transformation mit der Logit-Funktion durchgeführt. Der Logit bzw. das Chancenverhältnis ist das logarithmierte Verhältnis zwischen Eintritt und Nicht-Eintritt, definiert als die Wahrscheinlichkeit des Eintretens eines Ereignisses dividiert durch seine Gegenwahrscheinlichkeit[62]. Zwischen dem Logit und der Eintrittswahrscheinlichkeit besteht somit folgender Zusammenhang:

(4.12) $\textit{logit}\,(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \eta \;\Leftrightarrow\; p_i = \frac{1}{1+e^{-\eta}}$.

4.1.2.3 Schätzung der Koeffizienten

Ziel der logistischen Regression ist es, die Koeffizienten β_k derart zu schätzen, dass eine optimale Trennung der abhängigen Variablen erreicht wird. Für die Schätzung der Parameter wird für gewöhnlich die Maximum Likelihood-Methode[63] verwendet. Dafür wird nach Schätzwerten für die Parameter der zugrunde liegenden Verteilung gesucht, so dass die Wahrscheinlichkeitsmasse unterhalb der Dichte der Verteilung maximal wird. Folgende Log-Likelihood-Funktion ist somit gegeben[64]:

(4.13) $\sum_{i=1}^{n}\left(y_i \ln(p_i) + (1-y_i)\ln(1-p_i)\right) = \sum_{i|y=1}^{n_1} \ln(p_i) + \sum_{i|y=0}^{n_0}(1-p_i)$.

Die Bestimmung des Logits ist durch eine analytische Methode nicht zu erreichen, stattdessen bedarf es einer Umformung in ein Optimierungsproblem. Dies kann sehr aufwendig sein. Positiv bei Verwendung des Logits ist allerdings, dass es ein sehr gutes Optimierungsverfahren ist, da es sich um einen konvexen Lösungsraum handelt.

---

[62] Vgl. Backhaus, Erichson, Plinke, Weiber (2000), S. 109.

[63] Vgl. Winkler, (1997), S. 200 und Backhaus, Erichson, Plinke, Weiber (2000), S. 112.
Die Maximum Likelihood-Schätzmethode maximiert das Produkt der Zuordnungswahrscheinlichkeit zur jeweiligen Gruppe über alle Beobachtungen:
$L = \prod_{y_i=1} p(y_i = 1) \cdot \prod_{y_i=0} (1 - p(y_i = 1))$.

[64] n_0 und n_1 geben die Anzahl an Fälle für die Klassen 0 und 1 wieder.

4.1.3 Variablenauswahlverfahren

In der Regressionsanalyse existieren verschiedene Verfahren zur Auswahl der zur Verfügung stehenden Regressoren[65]. Die bekanntesten sind Forward Selection, Backward Selection und Stepwise Selection. Bei der Forward Selection wird diejenige Variable als erste in die Regression aufgenommen, welche die größte Korrelation mit der abhängigen Variablen aufweist. Jede weitere Aufnahme einer Variablen wird dann wiederum aufgrund der Wechselbeziehung zwischen Input- und Ziel-Variable entschieden. Die Backward Selection nimmt zuerst alle Variablen in das Modell auf und eliminiert danach Variablen auf der Basis von Signifikanz-Tests. Die Stepwise Selection ist eine Variante der Forward-Prozedur, wobei einzelne Variablen in späteren Schritten wieder eliminiert werden können.

4.2 Clusteranalyse

4.2.1 Einführung in die Clusteranalyse

Die Clusteranalyse als Klassifikationsverfahren mit den Forderungen der Homogenität der Objekte innerhalb der Cluster und der Heterogenität zwischen den Clustern verlangt nach den Eigenschaften disjunkte Klasseneinteilung (keine überlappenden Cluster), exhaustive Gruppierung (alle Objekte sind mindestens einer Gruppe zugeordnet) und einem Ähnlichkeitsmaß der Form

(4.14) $S_{ij} = S_{ji}$

und

(4.15) $S_{ij} \leq S_{ii} \; \forall \, i, j \in O,$

sowie einem Distanzmaß der Form

(4.16) $d_{ii} = 0, \; d_{ij} \geq 0 \; und \; d_{ij} = d_{ji} \; \forall \, i, j \in O,$

wobei O die Menge der zu klassifizierenden Objekte ist[66].

Eine Unterscheidung von Clusteranalyseverfahren wird anhand des Kriteriums der Gruppierungsform getroffen. Hierbei unterscheidet man zwischen hierarchischen und partitionierenden Verfahren. Bei partitionierenden Verfahren wird von einer vorgegebenen Gruppierung ausgegangen, die durch Austausch von Objekten zwischen den Klassen optimiert wird. Ein Objekt wird also i.d.R. im Laufe der Clusterung verschiedenen Gruppen zugeordnet. Bei hierarchischen Verfahren werden Klassen fusioniert oder unterteilt. Werden zu Beginn alle Objekte als eine Gruppe betrachtet und werden diese dann in kleinere unterteilt, spricht man von divisiven Verfahren. Bei dem agglomerativen

[65] Vgl. Stier (1999), S. 247.
[66] Vgl. Stier (1999). S. 323.

Verfahren wird zu Beginn jedes Objekt als eine eigene Gruppe angesehen. Durch iteratives, paarweises Zusammenfassen werden dann neue, größere Gruppen gebildet.

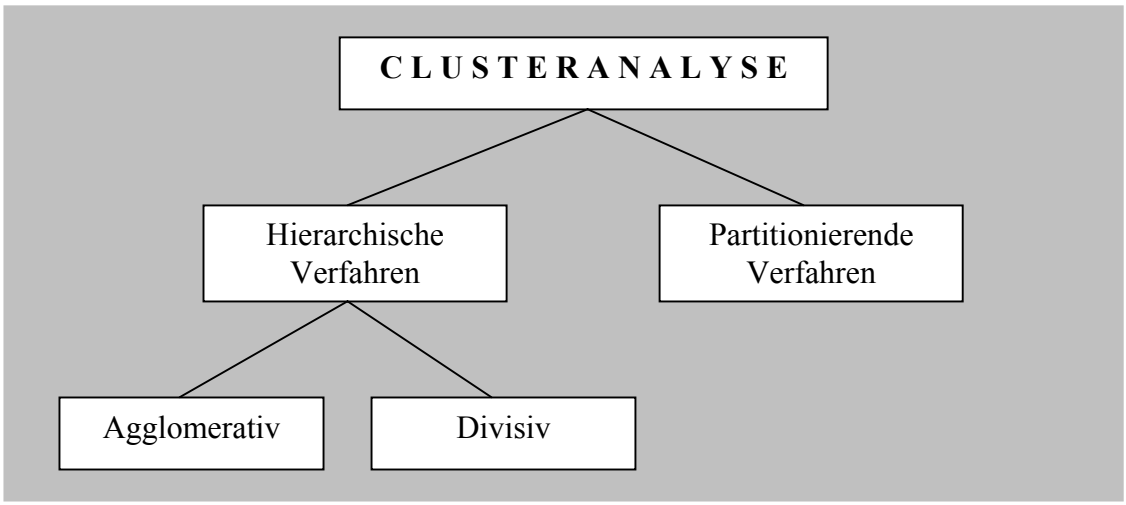


Abb. 4.4: Einteilung von Clusteranalyseverfahren.
Quelle: Eigene Darstellung.

4.2.2 K-Means-Verfahren

Bei den K-Means-Verfahren[67] werden Clusterzentren zur Bildung der Cluster konstruiert. Der Modellansatz besteht darin, dass die Clusterzentren für K Cluster so berechnet werden, dass die Streuungsquadratsumme in den Clustern minimiert wird.

(4.17) $$SQ_{in}(K) = \sum_{k} \sum_{g \in k} \sum_{j} (x_{gj} - \bar{x}_{kj})^2 \rightarrow \min .$$

Dies entspricht der quadrierten euklidischen Distanz $d^2_{g,k}$ zwischen dem Objekt g und dem Clusterzentrum k.

Die Minimierungsaufgabe aus (4.17) kann somit auch folgendermaßen formuliert werden:

(4.18) $$SQ_{in}(K) = \sum_{k} \sum_{g \in k} d^2_{g,k} \rightarrow \min .$$

Die Streuungsquadratsumme in den Clustern lässt sich als Fehlerstreuung interpretieren, also als Streuung in den Daten, die nicht durch die Cluster erklärt werden.

4.2.3 Der K-Means-Algorithmus[68]

Schritt 1: *Berechnung oder Eingabe von Startwerten für die Clusterzentren.*

Schritt 2: *Zuordnung der Klassifikationsobjekte:*

Die Klassifikationsobjekte g werden jenem Zentrum k zugeordnet, zu dem die quadrierte euklidische Distanz minimal ist. Dies führt dazu, dass die

[67] Vgl. Badner (1994), S. 308 f.
[68] Vgl. Badner (1994), S. 309 f.

Streuungsquadratsumme in den Clustern in jedem Iterationszyklus minimiert wird.

(4.19) $$SQ_{in}(K) = \sum_k \sum_{g \in k} d^2_{g,k} = \sum_g \min_{k^*=1,2,\ldots,K} (d^2_{g,k^*}).$$

Schritt 3: *Neuberechnung der Clusterzentren:*

Nach der Zuordnung aller Objekte zu den Clustern werden die Clusterzentren neu berechnet:

(4.20) $$\bar{x}_{kj} = \frac{\sum_{g \in k} x_{gj}}{n_{kj}}$$ [69].

Schritt 4: *Iteration:*

Es wird geprüft, ob sich im Schritt 2 die Zuordnung der Objekte geändert hat. Ist dies der Fall, werden die Schritte 2 und 3 erneut durchgeführt, ansonsten wird der Algorithmus beendet.

4.3 Entscheidungsbaumverfahren

4.3.1 Aufbau eines Entscheidungsbaumes

Standardgemäß wird bei der Entscheidungsbaumanpassung das Modell der rekurrierenden Partitionierung[70] verwendet. Die Anpassung erfolgt mit Hilfe des Top-Down-Verfahrens, dem „divide and conquer"-Prinzip („teile und herrsche") folgend. Außerdem wird das Verhalten bei der Auswahl der zu testenden Attribute als „greedy" („gierig") bezeichnet.

In einer gegebenen Menge von klassifizierten Fallbeschreibungen werden die bedingten Häufigkeitsverteilungen der Klassen unter den einzelnen zur Beschreibung verwendeten Attributen bestimmt und mit Hilfe eines Auswahlmaßes bewertet. Das Attribut mit der besten Bewertung wird als Testattribut ausgewählt. Dieser Teil des Algorithmus wird als „gierig" bezeichnet. Anschließend werden die Fallbeschreibungen gemäß der verschiedenen Werte des Testattributes aufgeteilt, und das Verfahren wird rekursiv auf die sich ergebenden Teilmengen angewandt. Dies ist der sog. „teile und herrsche"-Teil des Algorithmus. Die Rekursion bricht ab, wenn kein Attribut zu einer Verbesserung der Klassifikation führt oder keine weiteren Attribute für einen Test zur Verfügung stehen. Für den Aufbau eines Entscheidungsbaumes müssen Anzahl und Zulässigkeit[71] der Splits bestimmt werden, d.h. die Aufteilung der vorhandenen Daten anhand eines Klassifikators.

[69] n_{kj} gibt die Zahl der Objekte mit gültigen Angaben in der Variable j an, und auch nur solche Ausprägungen werden schließlich in die Summenbildung mit einbezogen.
[70] Top Down Induction of Decision Trees (TDIDT)
[71] Es sind lediglich univariate Splits zugelassen.

Als Standard werden binäre Bäume eingesetzt. Bei diesen Bäumen können die Daten nach einer Klassifikation in zwei Äste unterteilt werden. Eine andere Möglichkeit sind die sog. N-Way-Trees, bei denen mehrere Unterteilungen existieren, d.h. Verzweigungen.
Je nach Skalierung ergeben sich für die Anzahl der Splits unterschiedlich viele Aufteilungsmöglichkeiten. Bei ordinalen Inputvariablen handelt es sich um ordnungserhaltende Splits. Dies hat den Vorteil, dass die Aufteilungsmöglichkeiten verhältnismäßig gering sind. So ergeben sich für L verschiedene Ausprägungen und eine Anzahl von B möglichen Ästen $\binom{L-1}{B-1}$ Aufteilungen[72]. Splits mit nominalen Input-Variablen werden nicht mit Beschränkungen des Ordnungerhalts versehen, dadurch entsteht allerdings eine exponentiell wachsende Anzahl an Splitmöglichkeiten[73].

4.3.1.1 Algorithmen

Im SAS® Enterprise Miner™ ist keine Default-Einstellung für die Durchführung von Entscheidungsbaum-Algorithmen vorgesehen. Es ist aber möglich, eigenständig durch Bestimmung der Auswahlmaße (vgl. Abschnitt 4.3.1.2) und Anzahl der Splits die gängigsten Baumalgorithmen nachzubilden:

- CART (Classification and Regression Trees),
- CHAID (Chi-squared Automatic Interaction Detection) und
- ID3 (Iterative Dichotomiser 3) und dessen Nachfolger C4.5 und C5.0

4.3.1.2 Auswahlmaße

Die Induktion von Entscheidungsbäumen mit Hilfe eines Top-Down-Verfahrens ist eine bekannte und weit verbreitete Technik zur Bestimmung von Klassifikatoren. Der Erfolg dieser Methode hängt von dem Auswahlmaß[74] ab, mit dem beim Aufbau des Entscheidungsbaums das nächste zu testende Attribut bestimmt wird.

[72] Intervallskalierte Inputvariablen werden durch Berechnung der Klassifikation einer ordnungserhaltenden Transformation unterzogen und können damit wie ordinalskalierte Inputvariablen behandelt werden.
[73] Vgl. SAS Institute Inc. (2000a), S. 29 ff.
[74] Vgl. Nakhaeizadeh (1998), S. 80 ff.

4.3.1.2.1 Informationsgewinn und Entropie[75]

Das wohl bekannteste Auswahlmaß ist der Informationsgewinn (information gain). Er misst, wieviel Informationen man durch das Feststellen des Wertes des Testattributes über die Klasse gewinnt. Der hierbei verwendete Informationsbegriff basiert auf der Entropie H einer Wahrscheinlichkeitsverteilung:

(4.21) $H = -\sum_{i=1}^{n} \rho_{i.} \log_2 \rho_{i.}$ [76].

Der Informationsgewinn ist nichts anderes als die Entropieverminderung beim Übergang zur bedingten Verteilung und definiert als

(4.22) $I_{gain}(C, A) = H_C - H_{C|A} = H_C + H_A - H_{CA}$ [77]

$$= -\sum_{i=1}^{n_C} \rho_{i.} \log_2 \rho_{i.} - \sum_{j=1}^{n_A} \rho_{.j} \log_2 \rho_{.j} + \sum_{i=1}^{n_C} \sum_{j=1}^{n_A} \rho_{ij} \log_2 \rho_{ij}$$ [78].

4.3.1.2.2 Gini-Index

Der im vorhergehenden Abschnitt behandelte Informationsgewinn basiert auf der Entropie. Für das Lernen von Entscheidungsbäumen wird aber auch die quadratische Entropie verwendet:

(4.23) $H^2 = 2\sum_{i=1}^{n} \rho_{i.}(1-\rho_{i.}) = 2\left(1 - \sum_{i=1}^{n} \rho_{i.}^2\right)$.

Der Gini-Index[79] wird erreicht bei analogem Vorgehen zur Entropie:

(4.24) $Gini\ (C, A) = \frac{1}{2}\left(H_C^2 - H_{C|A}^2\right)$

$$= 1 - \sum_{i=1}^{n_C} \rho_{i.}^2 - \sum_{j=1}^{n_A} \rho_{.j}\left(1 - \sum_{i=1}^{n_C} \rho_{i|j}^2\right) = \sum_{j=1}^{n_A} \rho_{.j} \sum_{i=1}^{n_C} \rho_{i|j}^2 - \sum_{i=1}^{n_C} \rho_{i.}^2$$ [80].

Der Gini-Index kann somit als die zu erwartende Verringerung der Fehlerklassifikations-wahrscheinlichkeit gedeutet werden. Es wird angenommen, dass ein Fall zufällig klassifiziert wird, und zwar ρ_i als zur Klasse c_i gehörig. Dies wird offenbar mit Wahrscheinlichkeit $(1-\rho_i)$ falsch sein. Folglich gibt $\sum_{i=1}^{n} \rho_i(1-\rho_i)$ die Wahrscheinlichkeit an, dass ein Fall mit diesem Verfahren falsch klassifiziert wird. Indem man nun die zu

---

[75] Vgl. Nakhaeizadeh (1998), S. 85.
[76] $\rho_{i.}$ ist die relative Häufigkeit des Attributwertes c_i, $\rho_{.j}$ die relative Häufigkeit des Attributwertes a_j.
[77] H_C ist die Entropie der Klassenverteilung, H_A die Entropie der Attributwerteverteilung und H_{CA} die Entropie der gemeinsamen Verteilung.
[78] n_C steht für die Anzahl an Klassen, n_A für die Anzahl an Attributwerten.
[79] Vgl. Nakhaeizadeh (1998), S. 88.
[80] $\rho_{i|j}$ bezeichnet die relative Häufigkeit der Klasse c_i in den Fällen mit dem Attributwert a_j.

erwartende Fehlklassifikationswahrscheinlichkeit bei Kenntnis des Wertes des Attributes A abzieht, erhält man, analog zum Informationsgewinn, die Steigung der Wahrscheinlichkeit einer richtigen Klassifikation.

4.3.1.2.3 χ^2-Maß

Bisher wurde der Informationsgewinn als die zu erwartende Verringerung der Anzahl der Fragen, die zur Identifikation der wahren Klasse notwendig sind, gedeutet. Eine andere Formulierung lässt die Deutung zu, dass es ein Maß für den Unterschied der vorliegenden gemeinsamen Verteilung und der Verteilung ist, die sich bei Annahme der Unabhängigkeit der Attributwerte und der Klassen aus den Randverteilungen berechnen lässt. Für diesen Vergleich wird der Logarithmus Dualis der Verhältnisse einander entsprechender Wahrscheinlichkeiten der beiden Verteilungen summiert. Diese Form nennt man üblicherweise wechselseitige Information (mutual information).

$$(4.25)\quad I_{mutual}(C, A) = \sum_{i=1}^{n_C}\sum_{j=1}^{n_A} \rho_{ij} \log_2 \frac{\rho_{ij}}{\rho_{i.}\rho_{.j}}$$

$$= \sum_{i=1}^{n_C}\sum_{j=1}^{n_A} \rho_{ij} \log_2 \rho_{ij} - \sum_{i=1}^{n_C} \rho_{i.} \log_2 \rho_{i.} - \sum_{j=1}^{n_A} \rho_{.j} \log_2 \rho_{.j}$$

$$= - H_{CA} + H_C + H_A = I_{gain.}$$

Die wechselseitige Information ist also tatsächlich mit dem Informationsgewinn identisch. Die wechselseitige Information vergleicht die gemeinsame Verteilung mit einer hypothetischen, unabhängigen Verteilung mit Hilfe des Quotienten der Wahrscheinlichkeit. Ein Vergleich lässt sich aber durch Bildung des Abstandsquadrats durchführen. Dies führt zu dem χ^2-Maß[81]:

$$(4.26)\quad \chi^2(C, A) = \sum_{i=1}^{n_C}\sum_{j=1}^{n_A} \frac{\left(E_{ij} - N_{ij}\right)^2}{E_{ij}}, \text{ mit } E_{ij} = \frac{N_{i.}N_{.j}}{N_{..}}$$

$$= \sum_{i=1}^{n_C}\sum_{j=1}^{n_A} \frac{N_{..}^{\,2}\left(\frac{N_{i.}}{N_{..}}\frac{N_{.j}}{N_{..}} - \frac{N_{ij}}{N_{..}}\right)^2}{N_{..}\frac{N_{i.}}{N_{..}}\frac{N_{.j}}{N_{..}}}$$ [82]

$$= \sum_{i=1}^{n_C}\sum_{j=1}^{n_A} N \frac{\left(\rho_{i.}\rho_{.j} - \rho_{ij}\right)^2}{\rho_{i.}\rho_{.j}}.$$

[81] Vgl. Nakhaeizadeh (1998), S. 90.

[82] N.. steht für die Gesamtzahl der Fall- bzw. Objektbeschreibungen. $N_{i.}$ und $N_{.j}$ geben die absolute Häufigkeit der Klasse c_i bzw. des Attributwertes a_j an.

4.3.1.3 Stoppkriterien

Eine weitere Einschränkung bei der Modellierung von Entscheidungsbäumen ist neben der schon erwähnten Bestimmung der zulässigen Splits und der statistischen Signifikanz, d.h. der Mindestgröße der Auswahlmaße, die Tiefe des Baumes[83]. Die statistische Signifikanz wird durch den χ^2- bzw. den F-Test, den ein Split-Kriterium erfüllen muss, zu einer Stopp-Regel. Eine Verfeinerung dieser Verfahren wird mit der Baumtiefe durch den Tiefenmultiplikator erreicht. Dieser Multiplikator mit der Form $\prod_{i=0}^{d} B_i$ sorgt für eine Anpassung der P-Werte, die sich für die verschiedenen Tests ergeben.

4.3.2 Pruning

In den meisten Entscheidungsbaum-Lernprogrammen wird der konstruierte Baum anschließend gestutzt. Dies wird als Pruning[84] bezeichnet, d.h. einige Entscheidungsknoten, die nur geringen Anteil an der Klassifikationsgüte haben, werden wieder entfernt.

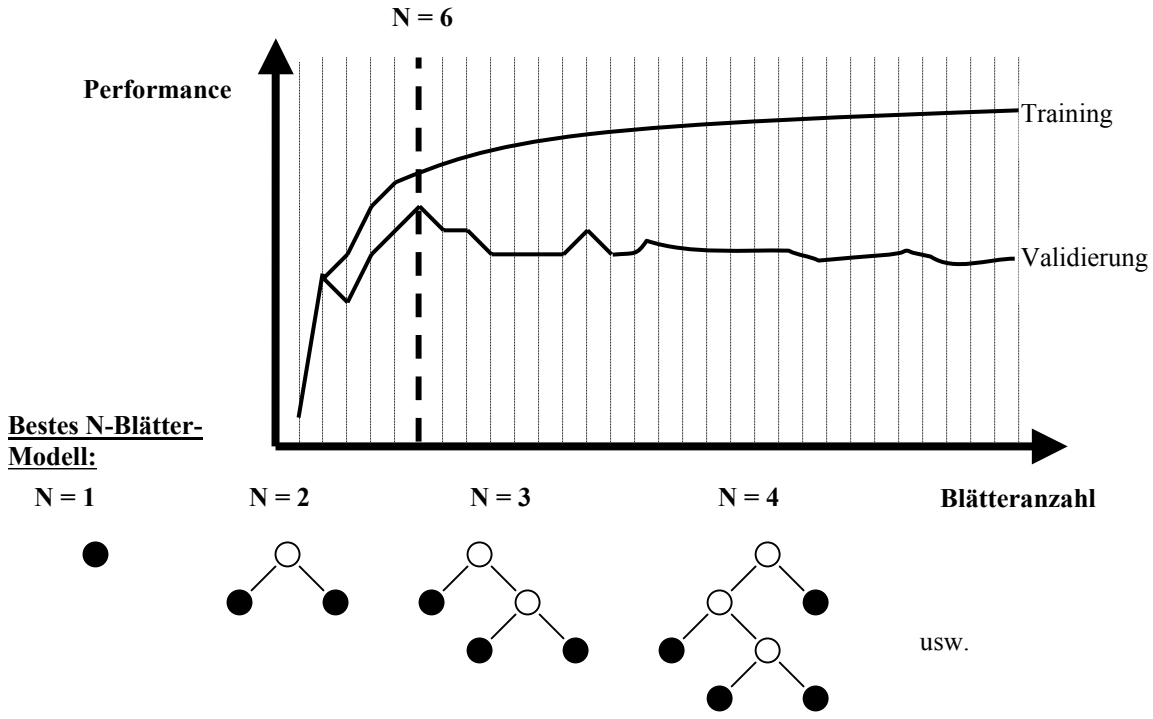


Abb. 4.5: Pruning eines Entscheidungsbaumes.
Quelle: SAS Institute Inc. (2000a), S. 70; eigene Darstellung.

[83] Vgl. SAS Institute Inc. (2000a), S. 68.
[84] Vgl. SAS Institute Inc. (2000a), S. 70.

Allgemein gilt: Je weiter der Baum aufgebaut wird, desto höher wird die Klassifikationsgüte. Allerdings verschlechtert sich ab einem gewissen Punkt die Generalisationsfähigkeit[85], die Gefahr des Overfittings droht. In Abbildung 4.5 ist unter bestes N-Blätter-Modell zu sehen, dass das beste Modell mit zwei Blättern nicht unbedingt die Grundlage für das beste Modell mit drei oder mehr Blättern bildet. Der Pruning-Prozess ist also so zu verstehen, dass, nachdem der Baum vollständig gebildet wurde, das Modell mit der besten Gesamtleistung ausgewählt wird. In Abbildung 4.5 wird das Modell mit sechs Blättern gewählt.
Dieses Verfahren wird auch Post-Pruning oder Pruning im engeren Sinne genannt, während die Stoppkriterien aus Abschnitt 4.3.1.3 als Pre-Pruning oder Pruning im weiteren Sinne gelten.

4.3.3 Surrogat-Splits für das Einfügen fehlender Werte

Fehlende Werte in den Trainingsdaten werden im Entscheidungsbaumverfahren als eigenständige Inputklasse betrachtet, daher sind geeignete Einfügestrategien nicht erforderlich. Dennoch können zur Modellanpassung und insbesondere für das Scoren neuer Fälle Surrogat-Splits verwendet werden. Ein Surrogat-Split nimmt eine Einteilung aufgrund der Nachahmung von anderen Inputs basierend auf den dafür ausgewählten Splits vor[86].

4.3.4 Wälder: Bagging und Boosting

Entscheidungsbäume sind instabile Modelle. Kleine Änderungen der Trainingsdaten können zu völlig veränderten Baumstrukturen führen. Die Instabilität resultiert aus der hohen Anzahl der denkbaren univariaten Splits (vgl. Abschnitt 4.3.1). Trotzdem bleibt die Gesamtleistung bezüglich der Klassifizierung der bekannten Daten hoch.
Eine weitere Schwachstelle der Entscheidungsbaumverfahren ist die in Abschnitt 4.0 erwähnte Schwierigkeit bei der linearen Separierbarkeit. Ein Modell, wie in Abbildung 4.6, wird durch wiederholte Stichprobenziehung und anschließenden Baumaufbau verschiedene Baumstrukturen liefern. Auch diese Eigenschaft verdeutlicht die Neigung zur Instabilität. Dieser Mangel der Entscheidungsbaumverfahren lässt sich jedoch auch dafür nutzen, mächtigere Modelle zu generieren. Indem man mit verschiedenen Stichproben der Trainingsdaten oder unterschiedlichen Methoden des Baumaufbaus operiert und die

[85] Vgl. Hastie, Tibshirani, Friedman (2001), S. 270.
[86] Vgl. SAS Institute Inc. (2000a), S. 61.

Resultate der dadurch entstehenden Bäume mittelt, lassen sich Mehrfach-Modelle[87] erzeugen. Dies wird auch als die P&C-Methode[88] bezeichnet. Diesbezüglich existieren verschiedene Ansätze, die bekanntesten sind Bagging und Boosting.

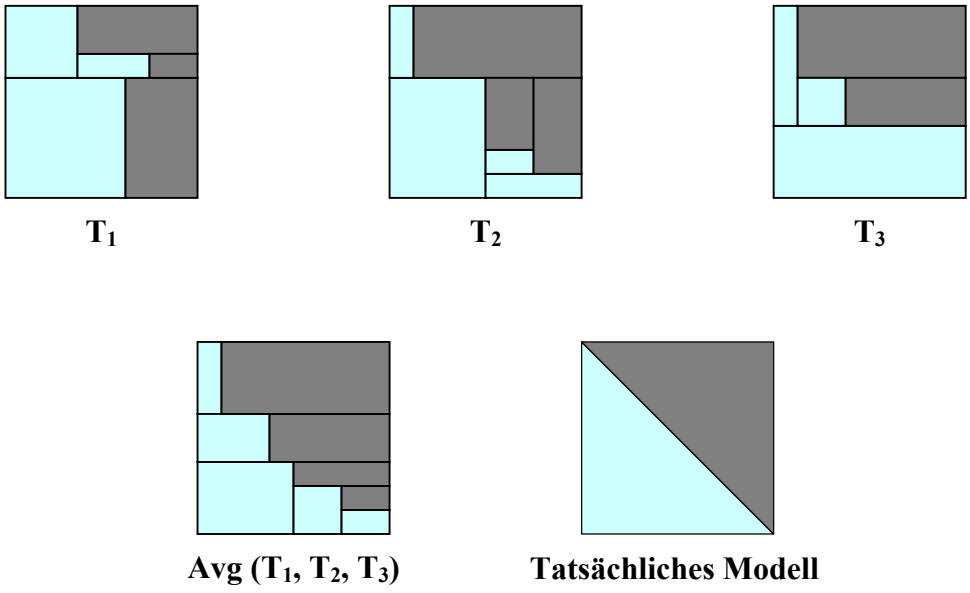


Abb. 4.6: Ensemble-Modell als Kombination von Mehrfach-Modellen.
Quelle: SAS Institute Inc. (2000a), S. 125; eigene Darstellung.

Bei Bagging-Verfahren[89] werden zuerst so viele Stichproben gezogen wie Durchläufe vorgesehen sind, danach werden auf Basis der Stichproben die Bäume aufgebaut. Idealerweise wird an dieser Stelle auf Pruning verzichtet, da große Bäume mit geringen Bias und hoher Varianz die besten Resultate liefern. Anschließend wird der Durchschnitt der Ereigniseintrittswahrscheinlichkeit gebildet[90].

Boosting-Verfahren[91] nehmen eine Störung, d.h. Veränderung, der Trainingsdaten vor, die auf den Ergebnissen der vorherigen Modelle basiert. Falsch klassifizierte Fälle werden in den folgenden Modellen stärker gewichtet. Für die Berechnung der Gewichte gilt:

$$(4.27)\quad p(i) = \frac{1 + m(i)^4}{\sum (1 + m(i)^4)}$$ [92].

Im Gegensatz zu Bagging mit Entscheidungsbäumen ist bei Boosting das Pruning von Vorteil.

[87] Jede zur Instabilität neigende Modellierungsmethode kann verwendet werden, wobei Entscheidungsbäume aufgrund ihrer Schnelligkeit und Flexibilität bevorzugt eingesetzt werden.
[88] Perturb and Combine-Modelle wenden das Konzept „Störung“ und „Kombination“ an.
[89] Vgl. Hastie, Tibshirani, Friedman (2001), S. 246 ff.
[90] Eine weitere Aggregierungsmöglichkeit ist die Mehrheitsbildung der vorhergesagten Fälle.
[91] Vgl. Hastie, Tibshirani, Friedman (2001), S. 299 ff.
[92] SAS Institute Inc. (2000a), S. 127.

4.4 Künstliche neuronale Netze

4.4.1 Einführung in die künstlichen neuronalen Netze

Neuronale Netze, oft auch als künstliche neuronale Netze (KNN) bezeichnet, sind informationsverarbeitende Systeme, die aus einer großen Anzahl einfacher Einheiten (Zellen, Neuronen) bestehen, welche sich Informationen in Form der Aktivierung der Zellen über gerichtete Verbindungen zusenden[93]. Das Studium neuronaler Netze ist motiviert durch die grobe Analogie zu den Gehirnen von Säugetieren, bei denen Informations-verarbeitung durch sehr viele Nervenzellen stattfindet, die im Verhältnis zum Gesamt-system sehr einfach sind und die den Grad ihrer Erregung über Nervenfasern an andere Nervenzellen weiterleiten. Das biologische Vorbild wird an dieser Stelle nur kurz vorgestellt, da es zum Verständnis der Arbeitsweise nicht unbedingt erforderlich ist. Für das Verständnis künstlicher neuronaler Netze reicht eine einfache Vorstellung über ein biologisches Neuron aus[94].

4.4.2 Netzwerkarchitektur

4.4.2.1 Multilayer Perceptron (MLP)

Zur Einführung in die Funktionsweise der KNN empfiehlt es sich, nochmals auf die multiple lineare Regression aus Abschnitt 4.1.1 sowie die logistische Regression aus Abschnitt 4.1.2 zu verweisen. Der Rahmen der künstlichen neuronalen Netze und insbesondere die Terminologie lassen sich anhand dieser statistischen Modelle gut erläutern. Eine Umwandlung der Regressionsgleichungen in ein Netzwerk-Diagramm zeigt wichtige Elemente der Netzwerkarchitektur: Eine Schicht (layer) mit Inputeinheiten ist mit einer Outputeinheit verbunden. Die Verbindungen (weights) sind gewichtet, wobei die Gewichte die Regressionskoeffizienten wiedergeben. Der konstante Term w_0 wird in dem Diagramm nicht repräsentiert, ist aber durch eine Verbindung zu einer Konstanten problemlos integrierbar. Die Outputeinheit der logistischen Regression wird mit der logit-Funktion transformiert. KNN werden üblicherweise in Netzwerk-Diagrammen[95] dargestellt, da die Schreibweise sehr umfangreich sein kann. Wie in den Netzwerk-Diagrammen aus Abbildung 4.7 besitzt das Multilayer Perceptron (MLP) ebenfalls eine Input- und Output-Schicht (vgl. Abb. 4.8). Die Input-Schicht enthält eine Einheit für jede Input-Variable. Die Output-Schicht repräsentiert das Ziel. Ein neues Element dieses

[93] Vgl. Zell (2000), S. 23 ff.
[94] Die Struktur eines biologischen Neurons wird im Anhang in Kapitel A.8 dargestellt.
[95] Vgl. SAS Institute Inc. (2000c), S. 26 ff.

Netzwerk-Diagramms für Feed-forward-Netze ist die sog. verdeckte Schicht, die Hidden-Layer, die verschiedene Einheiten, Neuronen, enthält. Die Inputs werden durch die Neuronen einem mathematische Transformationsprozess unterzogen. Dieses allgemeine Modell stellt also eine Transformation des erwarteten Ziels durch eine Linearkombination von nichtlinearen Funktionen dar, die wiederum die Inputs linear kombinieren.

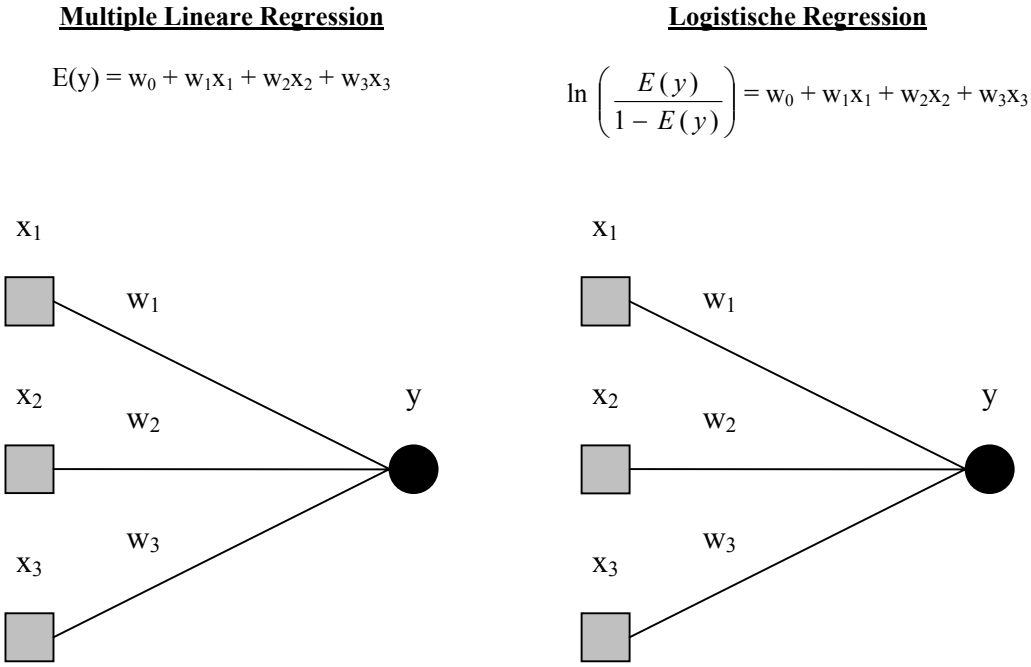


Abb. 4.7: Multiple lineare Regression und logistische Regression als Netzwerk-Diagramme.
Quelle: SAS Institute Inc. (2000c), S. 26; eigene Darstellung.

Bei der logistischen Regression wird mittels der logit-Funktion der Output transformiert, dieses lässt sich auch bei der multiplen linearen Regression beobachten. Allerdings ist der Transformator hier die Identität, so dass sich am Ergebnis nichts ändert. In diesem Zusammenhang wird von der Output-Aktivierungsfunktion[96] bzw. der Link-Funktion gesprochen. Die Transformation des erwarteten Ziels der MLP-Architektur ist die Inverse der Output-Aktivierungsfunktion $g_0()$. Verschiedene Link-Funktionen stehen zur Auswahl: Für unbeschränkte intervallskalierte Zielvariablen wird die Identität verwendet. Bei einer Nicht-Negativitäts-Restriktion ist die Log-Link-Funktion gebräuchlich, die den Output exponentiell darstellt. Liegen die Zielwerte zwischen Null und Eins, wird eine logistische Transformation vorgenommen (Logit-Link-Funktion). Weitere Link-Funktionen sind der generalisierte sowie der kumulative Logit, die durch die Softmax- bzw. die logistische Funktion polychotome und ordinale Zielwerte ermöglichen. Bei polychotomen bzw. ordinalen Zielen sind zwei Output-Einheiten angelegt. Weitere Aktivierungsfunktionen

[96] Vgl. Zell (2000), S. 72 ff.

besitzt jedes Neuron der Hidden-Layer. Das Argument einer Aktivierungsfunktion nennt man Kombinationsfunktion und ist in einer MLP eine Linearkombination[97].

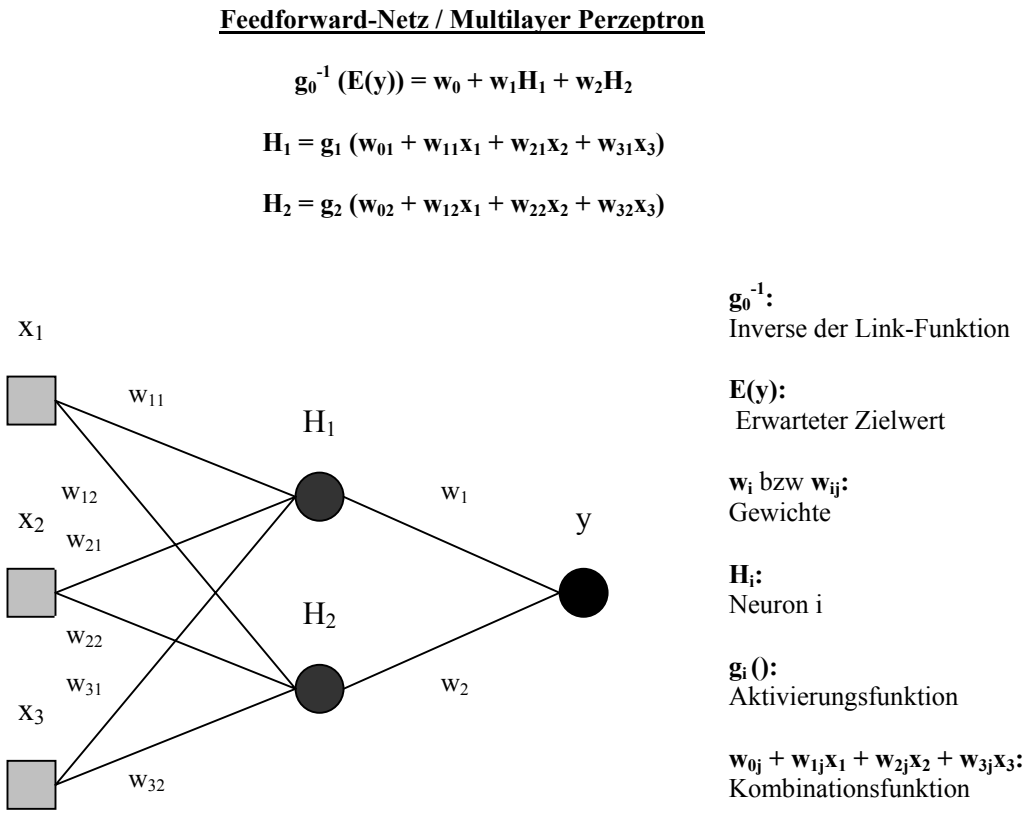


Abb. 4.8: Netzwerk-Diagramm des Feedforward-Netzes bzw. des Multilayer Perzeptrons.
Quelle: SAS Institute Inc. (2000c), S.27; eigene Darstellung.

Die Aktivierungsfunktionen der Neuronen bewirken die nicht-lineare Transformation in neuronalen Netzwerken. Verschiedene Funktionen mit einem sigmoiden Verlauf können verwendet werden. Üblicherweise sind das der Tangens Hyperbolicus oder die logistische Funktion[98].

Unterschiede zwischen diesen Funktionen[99] sind lediglich der Gültigkeitsbereich und die Steigung, was folgende Reparametrisierung zeigt:

$$(4.28)\quad tanh(\eta) = \frac{e^{\eta} - e^{-\eta}}{e^{\eta} + e^{-\eta}} = \frac{1 - e^{-2\eta}}{1 + e^{-2\eta}} = \frac{2}{1 + e^{-2\eta}} - 1 = 2\ logisitic\ (2\eta) - 1.$$

Die bisherigen Erklärungen zu dem Element Neuron und der dazugehörigen Aktivierungsfunktion wird ergänzt durch den Aktivierungszustand eines Neurons. Der Aktivierungszustand $a_i(t)$ gibt den Grad der Aktivierung an. In der technischen Realisierung existieren verschiedene Darstellungsformen. Es werden kontinuierliche und diskrete Wertebereiche unterschieden. Bei kontinuierlichen Wertebereichen werden

[97] Radiale-Basisfunktionen-Netze (Abschnitt 4.4.2.2) verwenden keine Linearkombination.
[98] Weitere sigmoide Aktivierungsfunktionen sind die Elliott-Funktion und der Arcus Tangens.
[99] Die logistische Funktion und Tangens Hyperbolicus werden im Anhang in Kapitel A.9 abgebildet.

entweder alle reelen Zahlen als Werte zugelassen oder es erfolgt eine Beschränkung bei die Aktivierung auf ein Intervall: [0, 1] bzw. [-1, 1]. Eine Intervallaktivierung ist der Regelfall. Aufgrund theoretischer Modelle sind auch diskrete Aktivierungszustände möglich, d.h. bei der Implementierung werden lediglich binäre Werte, z.B. {0, 1}, zugelassen.

Durch den Aktivierungszustand lässt sich wiederum die Aktivierungsfunktion präzisieren[100]. Die Aktivierungsfunktion f_{act} gibt demnach an, wie sich ein neuer Aktivierungszustand $a_j(t+1)$ des Neurons j aus der alten Aktivierung $a_j(t)$ und der Netzeingabe, dem net input ($net_j(t)$), bestimmt:

(4.29) $a_j(t+1) = f_{act}\,(a_j(t),\, net_j\,(t),\, \Phi_j)$,

wobei Φ_j der Schwellenwert der Funktion des Neurons j ist.

Schließlich wird die Ausgabe der Zelle j durch eine Ausgabefunktion aus der Aktivierung der Zelle bestimmt:

(4.30) $o_j = f_{out}\,(a_j)$.

Die Verbindungen, die sog. Gewichte, stehen für die im Modell verwendeten Koeffizienten der verschiedenen Terme. Die Gewichte sind unbekannte Parameter, die bei der Modellanpassung an die Daten geschätzt werden. Der verbleibende Bias des Modells wird entweder durch eine Konstante repräsentiert oder ausgeblendet. Für die Parameteranzahl n in einer MLP-Architektur mit einer Hidden-Layer mit h Neuronen und k Inputs gilt:

(4.31) $n = h(k+1) + h + 1 = h(k + 2) + 1$.

Die letzte Funktion, die im Zusammenhang mit Neuronen erwähnt werden muss, ist die Propagierungsfunktion. Sie gibt an, wie sich die Netzeingabe eines Neurons aus den Ausgaben der anderen Neuronen und den Verbindungsgewichten berechnet. Die Netzeingabe $net_j(t)$ von Zelle j berechnet sich nach

(4.32) $$net_j(t) = \sum_{i=1}^{N} o_i(t) w_{ij}$$

aus der Summe der Ausgaben $o_i(t)$ der Vorgängerzellen multipliziert mit dem jeweiligen Gewicht w_{ij} der Verbindungen von Zelle i nach Zelle j.

Das in Abbildung 4.8 gezeigte MLP-Netzwerk kann, neben Änderungen bezüglich der Input- oder der Neuronenanzahl in der Hidden Layer, weiter modifiziert werden. Zu den schon vorhandenen, gerichteten Kanten sind sog. Shortcut Connections möglich, also direkte Verbindungen von den Input-Einheiten zu der Output-Einheit[101].

[100] Vgl. Zell, A. (2000), S. 72 ff.
[101] Verschiedene Topologien KNN sind im Anhang in Kapitel A.10 zu sehen.

Veränderungen bei KNN können durch weitere verdeckte Schichten vorgenommen werden. Theoretisch ist das nicht nötig, da KNN bei genügender Anzahl an Neuronen als universelle Approximatoren fungieren. Durch eine Veränderung der Gewichte sowie deren Vorzeichen können verschiedenste Flächen abgedeckt werden. Die Neuronenanzahl lässt sich allerdings nicht extern optimieren, so dass die Gefahren Under- bzw. Overfitting drohen (vgl. Kapitel 4, Vorbemerkungen). Eine weitere verdeckte Schicht ist leider auch kein Garant für ein gutes Modell – die Grundproblematik bleibt auch hier erhalten. Bei stark zerklüfteten Räumen kann eine zusätzliche Hidden Layer positive Ergebnisse bringen. Eine gute Modellabstimmung kann aber auch hier nur über verschiedene Anpassungsschritte erfolgen.

An dieser Stelle sollte noch kurz auf den Begriff Multilayer Perceptron eingegangen werden. Ursprünglich wurden unter dem Begriff Perceptron einfache vorwärts gerichtete Netze verstanden, die keine innere Schicht besitzen. Dadurch ist allerdings die Leistungsfähigkeit stark eingeschränkt. Die Erweiterung auf mehrstufige Perceptrons mit mehreren trainierbaren Schichten bewirkt die Erfüllung des Konvergenz-Theorems, das fordert, dass jede repräsentierbare Funktion in endlicher Zeit antrainiert werden kann.

Die letzte Modifikation ist die Veränderung der Kombinationsfunktion. Anstatt der bisherigen Linearkombination nutzt die im nächsten Abschnitt besprochene Netzwerkarchitektur der sog. Radialen-Basisfunktionen-Netze eine radialsymmetrische Kombinationsfunktion.

4.4.2.2 Radiale-Basisfunktionen-Netze (RBF-Netze)

Radiale-Basisfunktionen-Netze[102] sind vorwärtsgerichtete KNN, die nur eine Schicht verdeckter Neuronen besitzen. Charakteristisch für RBF-Netze[103] ist die radialsymmetrische Kombinationsfunktion. Die Neuronen der verdeckten Schicht werden mit Glockenform erzeugender (Gauss) Oberfläche zentriert im Eingaberaum positioniert. Die Kombinationsfunktion berechnet nun den Abstand zwischen jedem Datenpunkt und dem Zentrum[104]. Diese Stützstellen werden durch Gewichte bestimmt $(x_1, x_2, \ldots, x_k) = (w_{1i}, w_{2i}, \ldots, w_{ki})$. Der Term w_{0i} beeinflusst die Breite der Basis-Funktionen. Das Vorzeichen der Gewichte, die von der verdeckten Schicht zur Outputschicht wirken, bestimmen, ob die Oberfläche konvex oder konkav ist.

[102] Vgl. Zell (2000), S. 225 ff.
[103] Vgl. SAS Institute Inc. (2000c), S. 65 ff.
[104] Mit Hilfe der Werte dieser Stützstellen werden mehrdimensionale Funktionen approximiert.

Aus Gründen der Übersichtlichkeit wird in den nun folgenden Netzwerk-Diagrammen auf die Verwendung von Bezeichnungen für die einzelnen Gewichte, die Input- und Output-Einheiten sowie die Neuronen der verdeckten Schicht verzichtet.

Radiale-Basisfunktionen-Netze

$$g_0^{-1}(E(y)) = w_0 + w_1H_1 + w_2H_2$$

$$H_1 = \exp\left(-w_{01}^2((x_1 - w_{11})^2 + (x_2 - w_{21})^2 + (x_3 - w_{31})^2)\right)$$

$$H_2 = \exp\left(-w_{02}^2((x_1 - w_{12})^2 + (x_2 - w_{22})^2 + (x_3 - w_{32})^2)\right)$$

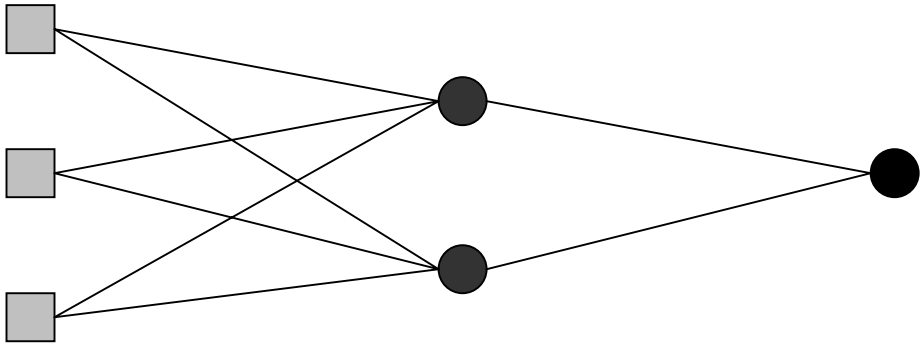


Abb. 4.9: Radiale-Basisfunktionen-Netze.
Quelle: SAS Institute Inc. (2000c), S. 65; eigene Darstellung.

Diese allgemeine Darstellungsform kann durch eine Normalisierung noch weiter spezifiziert werden. Die Normalisierten Radiale-Basisfunktionen-Netze (NRBF-Netze) verwenden eine Softmax[105] Aktivierungsfunktion. Die Basisfunktionen stellt somit das Verhältnis einer Gauss'schen Oberfläche zu einer weiteren Gauss'schen Oberfläche dar. Die Bedingung der Softmax-Funktion bewirkt einen Verteilungseffekt[106] bei den Basis-Funktionen. NRBF-Netze besitzen einen Höhen-Parameter[107] a_i (vgl. Abb. 4.10), der die Maximalhöhe angibt. Der Höhen-Parameter ermöglicht den einzelnen Gauss'schen-Funktionen neben verschiedenen Breiten, erzeugt durch die Gewichte, auch verschiedene Höhen. Die Konstante f gibt die Anzahl der Verbindungen zu dem Neuron an.

[105] Die einzelnen Terme einer Softmax-Funktionen addieren sich zu Eins: $H_1 + H_2 + \ldots + H_n = 1$.
[106] MLP-Netze besitzen ebenfalls durch ihren sigmoiden Verlauf einen Verteilungseffekt. Jede Basis-Funktion eines NRBF-Netzes liefert Ergebnisse zwischen 0 und 1 ($0 \leq H_i \leq 1$). Die Bedingung der Softmax-Funktion und der damit verbundene Verteilungseffekt verschaffen den NRBF-Netzen eine hohe Flexibilität.
[107] In den ursprünglichen RBF-Netzen ist der Höhen-Parameter redundant zu den Gewichten, die von der verdeckten Schicht zur Outputschicht wirken.

Daraus ergibt sich für NRBF-Netze folgendes Netzwerk-Diagramm:

NRBF-Netze

$$g_0^{-1}(E(y)) = w_1H_1 + w_2H_2 + w_3H_3$$

$$H_1 = \frac{e_1}{e_1 + e_2 + e_3} \qquad H_2 = \frac{e_2}{e_1 + e_2 + e_3} \qquad H_3 = \frac{e_3}{e_1 + e_2 + e_3}$$

$$e_i = \exp\left(f \cdot \ln(a_i) - w_{0i}^2\left((x_1 - w_{1i})^2 + (x_2 - w_{2i})^2 + (x_3 - w_{3i})^2\right)\right)$$

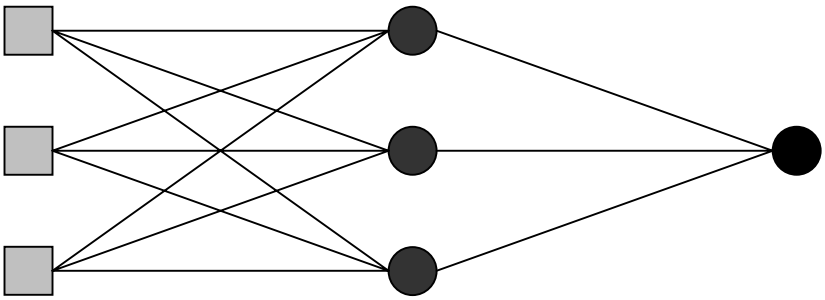


Abb. 4.10: Normalisierte Radiale-Basisfunktionen-Netze.
Quelle: SAS Institute Inc. (2000c), S. 68; eigene Darstellung.

Bezüglich der Breite und der Höhe lassen sich folgende Bedingungen aufstellen:

Varianten des NRBF-Netzes

Varianten	**Kurvenverläufe**	**Parameteranzahl**
1. Gleiche Breite, gleiche Höhe Equal widths and heights		h (k + 1) + 1
2. Unterschiedliche Breite, gleiche Höhe Unequal widths and equal heights		h (k + 2)
3. Gleiche Breite, unterschiedliche Höhe Equal widths and unequal heights		h (k + 2) + 1
4. Gleiches Volumen Equal volume		h (k + 2)
5. Unbestimmte Verläufe Unconstrained		h (k + 3)

Abb. 4.11: Einschränkungen bezüglich der Gestaltung des Höhen-Parameters und der Gewichte.
Quelle: SAS Institute Inc. (2000c), S. 69; eigene Darstellung.

Durch die in Abbildung 4.11 beschriebenen Einschränkungen lassen sich bezüglich des Höhen-Parameters und der Gewichte bei den einzelnen Gauss'schen Funktionen verschiedene Modifikationen für NRBF-Netze vornehmen. Generell gilt: Je freier der Verlauf, desto geringer ist die Parametrisierung des NRBF-Netzes.

4.4.3 Lernregel

Die Lernregel ist ein Algorithmus, gemäß dem das neuronale Netz lernt, für eine vorgegebene Eingabe eine gewünschte Ausgabe zu produzieren[108]. Lernen[109] erfolgt in neuronalen Netzen meist durch Modifikation der Verbindungsstärke als Ergebnis der wiederholten Präsentation von Trainingsmustern. Oft wird dabei versucht, den Fehler zwischen erwarteter Ausgabe und tatsächlicher Ausgabe zu minimieren. Wenn man den Fehler eines neuronalen Netzes als Funktion der Gewichte des Netzwerks grafisch aufträgt, erhält man eine Fehlerfläche[110], die sich im zweidimensionalen Fall darstellen lässt (vgl. Abbildung 4.12). Als Fehlerfunktion wird häufig der quadratische Fehler zwischen erwarteter und realer Ausgabe verwendet[111]. Der Gesamtfehler E[112] ergibt sich als Summe der Fehler über alle Muster p, einschließlich des Faktors 1/2[113].

(5.33) $$E = \sum_p E_p \; mit \; E_p = \sum_j \left(t_{pj} - o_{pj}\right)^2 .$$

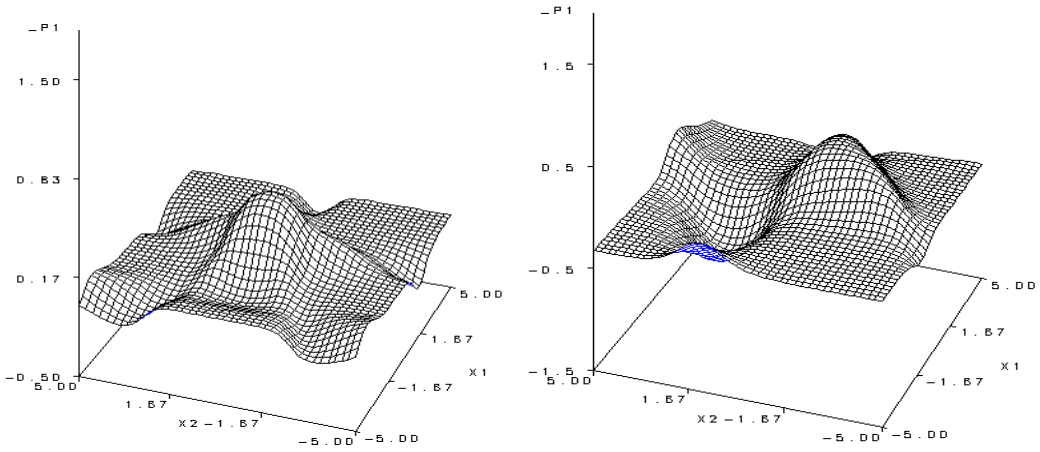


Abb. 4.12: Mögliche Fehlerflächen eines neuronalen Netzes als Funktion der Gewichte w_1 und w_2.
Quelle: Screenshot SAS® System Help; eigene Darstellung.

Es können verschiedene Zielfunktionskriterien, wie die absolute Abweichung oder die Cross Entropy verwendet werden. Häufig basieren die Fehlerfunktionen auf dem Maximum Likelihood-Prinzip, wobei allerdings aus rechnerischer Vereinfachung das Negative der logarithmierten Likelihood-Funktion minimiert wird.

---

[108] Vgl. Zell. (2000), S. 93.
[109] Das Lernen bzw. das Training entspricht der Schätzung der unbekannten Parameter in einem KNN. Alle Punkte in einem p-dimensionalen Parameterraum sind mögliche Lösungen.
[110] Vgl. Lämmel, Cleve (2001), S. 192.
[111] Vgl. Benenati (1998), S. 15.
[112] Dabei ist E_p der Fehler für ein Muster p, t_{pj} die Lerneingabe, o_{pj} die Ausgabe von Neuron j bei Muster p.
[113] Der Faktor 1/2 wird verwendet, damit er sich nach der Differenzierung mit der dadurch entstehenden 2 wegkürzt. Zur Bestimmung des Optimums ist es unerheblich, ob der Fehler oder der halbe Fehler minimiert wird. Gleiches gilt für die Verwendung des Quadrats des euklidischen Abstandes.

4.4.3.1 Gradientenabstiegsverfahren

Alle Gradientenverfahren berechnen den Gradienten einer Zielfunktion und steigen entweder orthogonal nach oben, bis ein Maximum erreicht ist, oder analog nach unten zu einem Minimum. Das Gradientenabstiegsverfahren versucht, die Fehlerminimierung über die Änderung der Gewichte zu realisieren. In allen Gewichten wird eine Änderung um einen Bruchteil des negativen Gradienten der Fehlerfunktion[114] vorgenommen.

(4.34) $\Delta W = -\eta \nabla E(W)$.

Die Änderung des Gewichtsvektors ΔW ist proportional zum negativen Gradienten $-\nabla E(W)$ der Fehlerfunktion mit dem Faktor η[115].

4.4.3.1.1 Probleme bei Gradientenverfahren

Die Gradientenverfahren besitzen eine Reihe von Problemen, die dadurch entstehen, dass es sich um lokale Verfahren handelt, bei denen Informationen über die Fehlerfläche nicht vorhanden sind[116]. Es besteht lediglich Kenntnis über die lokale Umgebung.

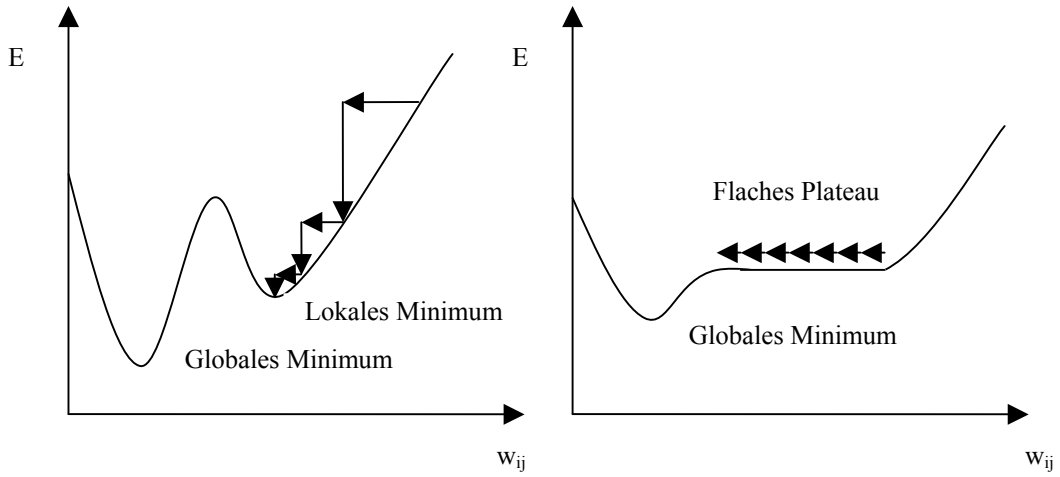


Abb. 4.13: Lokales Minimum einer Fehlerfläche und Fehlerfläche mit weiten Plateaus.
Quelle: Zell (2000), S. 113; eigene Darstellung.

Gradientenverfahren haben alle das Problem, dass sie in einem lokalen Minimum der Fehlerfläche konvergieren können. Die Wahrscheinlichkeit des Erreichens eines solchen suboptimalen Minimums steigt bei wachsender Dimension und damit verbunden einer stärker zerklüfteten Oberfläche. Ein weiteres Problem von Gradientenverfahren sind flache Plateaus, bei denen das Lernverfahren eine extrem hohe Anzahl an Iterationsschritten benötigt. Bei vollständig flachen Plateaus ist der Gradient der Nullvektor, d.h. es werden keine weiteren Gewichtsänderungen mehr durchgeführt. Des Weiteren ist nicht ersichtlich,

[114] Vgl. Zell (2000), S. 106.
[115] Der Faktor η wird als Lernfaktor oder Schrittweise bezeichnet.
[116] Vgl. Zell (2000), S. 113.

ob es sich um eine Stagnation in einem Plateau oder um ein Minimum handelt, bei dem der Gradient ebenfalls der Nullvektor ist.

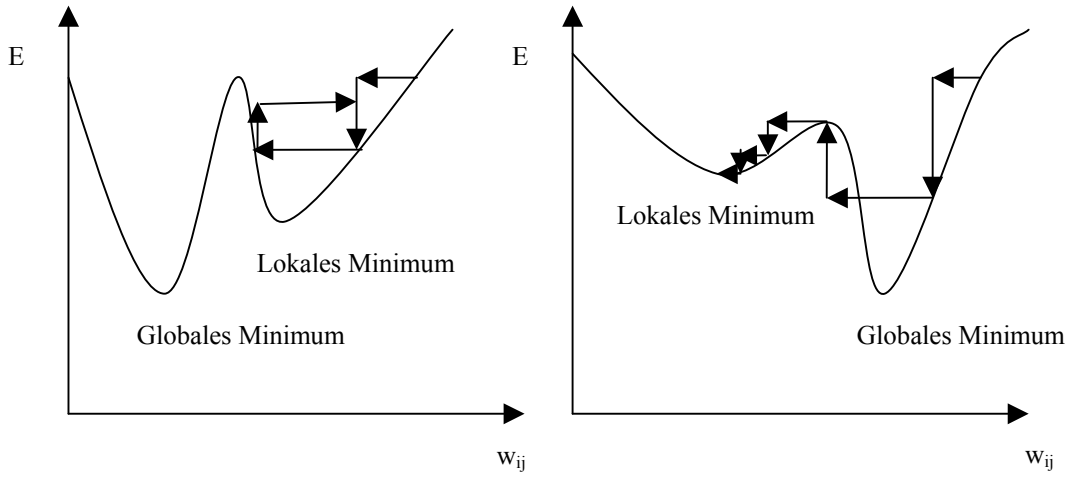


Abb. 4.14: Oszillationen in steilen Schluchten und Verlassen guter Minima.
Quelle: Zell (2000), S. 113; eigene Darstellung.

In steilen Schluchten der Fehlerfläche kann das Lernverfahren oszillieren. Ab einer gewissen Gradientengröße bewirkt eine Gewichtsänderung einen Sprung auf die andere Seite der Schlucht. Bei entsprechender Steilheit der anderen Seite kommt es zur Oszillation. Besitzt die Fehlerfläche enge Täler mit entsprechend hohen Gradienten, kann es zu einem Verlassen guter Minima kommen. Eine Gewichtsänderung kann in diesem Fall zu einem suboptimalen Minimum führen.

4.4.3.2 Backpropagation

Backpropagation[117] ist ein Lernverfahren für Netze mit n Schichten trainierbarer Gewichte und für Neuronen mit einer nichtlinearen Aktivierungsfunktion. Der Ausgangspunkt ist das Gradientenverfahren, das – als einzelnes Argument dargestellt – folgende Form hat:

$$(4.35)\quad \Delta w_{ij} = -\eta \frac{\partial}{\partial w_{ij}} E(W) = \sum_p -\eta \frac{\partial}{\partial w_{ij}} E_p \ .$$

Dies ist der Anfang für die Herleitung der Backpropagation-Regel[118]. Die Regel für Backpropagation lautet schließlich:

$$(4.36)\quad \Delta w_{ij} = \eta o_j (t_j - a_j) = \eta o_j \delta_j$$

$$\text{mit } \delta_j = \begin{cases} o_j(1-o_j)(t_j - o_j) & \text{falls j eine Ausgabezelle ist.} \\ o_j(1-o_j)\sum_k (\delta_k w_{jk}) & \text{falls j eine verdeckte Zelle ist.} \end{cases}$$

[117] Vgl. Zell (2000), S. 105 ff.
[118] Die Vorgehensweise für die Herleitung der Backpropagation-Regel ist im Anhang in Kapitel A.11 nachzuvollziehen.

Die Gewichtsänderung ist somit proportional zur Differenz δ_j der aktuellen Aktivierung a_j und der erwarteten Aktivierung t_j. Das seit 1986 existierende Bachpropagation-Verfahren hat über die Jahre verschiedene Modifikationen erfahren:

- Backpropagation mit Momentum[119]

Es wird zusätzlich die vorherige Gewichtsänderung $\Delta w_{ij}(t-1)$ zum Zeitpunkt t-1 mit einbezogen:

(4.37) $\Delta w_{ij}(t) = \eta o_i \delta_j + \mu \Delta w_{ij}(t-1)$.

Der Faktor μ (im Bereich zwischen 0 und 1) steuert den Anteil dieser Veränderung.

- QuickProp

Das QuickProp-Verfahren greift bei der Bestimmung der Gewichtsänderung auf die Annahme zurück, dass sich die Fehlerfunktion durch eine quadratische Funktion besser lokal approximieren lässt und somit der optimale Wert für ein Gewicht schneller gefunden werden kann. Zur Bestimmung der quadratischen Funktion werden im Quickprop-Verfahren[120] die letzten beiden Gewichtswerte $w_{ij}(t-1)$ und $w_{ij}(t-2)$ herangezogen.

- Resilient Propagation (RPROP)[121]

Resilient heißt federnd. Das Verfahren ändert die Gewichte nicht entsprechend dem Betrag des Gradienten, sondern nutzt nur sein Vorzeichen. Dazu wird das Vorzeichen des Kurvenanstiegs zum Zeitpunkt t sowie zum vorherigen Zeitpunkt herangezogen. Vorzeichen und Betrag werden getrennt bestimmt. Haben die beiden letzten Anstiege dieselbe Richtung, wird eine betragsmäßig höhere Veränderung in diese Richtung vorgenommen. Waren die Anstiege von unterschiedlichen Vorzeichen, wird die letzte Änderung zurückgenommen und das Gewicht mit kleineren Beträgen verändert.

- Backpercolation

Dieses Verfahren der sog. Untertunnelung liefert für viele Anwendungen die schnellste Anpassung der Gewichte. Es wurde insbesondere für Netze mit mehreren inneren Schichten entwickelt. Die Gewichtsänderungen beim Backpropagation-Algorithmus werden von Schritt zu Schritt kleiner, so dass die Anpassung der ersten Schicht erschwert wird. Backpercolation[122] greift hier ein und ermöglicht auch für die vorderen inneren Schichten höhere Änderungsraten der Gewichte und beschleunigt somit den Anpassungsprozess.

---

[119] Vgl. Zell (2000), S. 115.
[120] Vgl. Lämmel, Cleve (2001), S. 203 f.
[121] Vgl. Lämmel, Cleve (2001), S. 204.
[122] Vgl. Benenati (1998), S. 23 f.

4.4.3.3 Konjugierter Gradientenabstieg

Eine deutliche Verbesserung des einfachen Gradientenabstiegs erreichen konjugierte Gradientenabstiegsverfahren[123], die außer der Richtung des aktuellen Gradienten noch die vergangene Abstiegsrichtung mit einbeziehen.

4.4.3.4 Newton-Verfahren

Es werden drei verschiedene Newton-Verfahren[124] unterschieden: Newton-Raphson, Quasi-Newton und Gauss-Newton. Alle Verfahren verwenden die Approximation zur Minimierung, sei es der Zielfunktion (Newton-Raphson) oder der Hessematrix (Quasi-Newton und Gauss-Newton), deren Berechnung bei jedem Iterationsschritt erforderlich ist.

4.4.3.5 Levenberg-Marquard

Levenberg-Marquard[125] ist kein eigenständiges Optimierungsverfahren, sondern wird zur Modifikation von bestehenden Verfahren eingesetzt, beispielsweise Gauss-Newton oder Newton-Raphson. Die Levenberg-Marquard-Modifikation bietet Verbesserungen der Konvergenz insbesondere dann, wenn sich die Zielfunktionen nur schlecht quadratisch approximieren lassen.

4.4.4 Regulierbarkeit

4.4.4.1 Early Stopping

Neuronale Netzwerke approximieren Funktionen, jedoch mit der Einschränkung, dass diese von Störtermen überlagert werden. In der Trainingsphase werden die vorhandenen Trainingsdaten ausschließlich zur Bestimmung der Gewichte, also zum Schätzen der Parameter, verwendet. Ab einem gewissen Zeitpunkt des Lernprozesses beginnt das KNN, die Störterme zu approximieren (Overfitting). Das Ziel ist jedoch nicht nur, gute Resultate in Bezug auf die Trainingsdaten zu liefern, sondern auch gut zu generalisieren. Diese Fähigkeit geht verloren, wenn sich die Netzwerkgewichte zu stark an die Trainingsdaten anpassen. Early Stopping[126] oder Stopped Training bricht das Training ab, wenn der Fehler der Validierungsmenge sein Minimum erreicht hat. Dieser fängt nämlich genau dann an zu steigen, wenn das Netz beginnt, die Störterme zu approximieren. Das Problematische an diesem Verfahren ist, dass lediglich die Symptome behandelt werden. Overfitting hat seine

[123] Vgl. Bishop (1995), S. 275 ff.
[124] Vgl. Anders (1995), S. 19 ff.
[125] Vgl. Anders (1995), S. 23.
[126] Vgl. SAS Institute Inc. (2000c), S. 182 ff.

Ursache in der Überparametrisierung, also in einem zu umfassend spezifizierten Modell, welches auch nach dem Early Stopping immer noch zu komplex ist.
Early Stopping wird im SAS® Enterprise Miner™ automatisch verwendet und sorgt so dafür, dass die Fähigkeit zur Generalisierung gewahrt bleibt.

4.4.4.2 Weight Decay

Weight Decay[127], d.h. Gewichtsabnahme, wirkt großen Gewichten entgegen, da dies zu einer steilen und zerklüfteten Fehlerfläche[128] führt. Dies verlangt nach einem kleinen Betrag der Gewichte bei gleichzeitiger Annäherung an die Zielvorgaben der Trainingsmenge. Weight Decay findet seine Umsetzung in einer veränderten Fehlerfunktion. Die neue Zielfunktion addiert zur ursprünglichen Fehlerfunktion einen Strafterm hinzu:

(4.38) $E = E_0 + \lambda W^2 \; mit \; W = \sum_{i,j} w_{ij}$.

Der Parameter λ steuert das Ausmaß der Strafe.
Im Gegensatz zu Early Stopping wird Weight Decay nicht automatisch verwendet. Bei der Implementierung dieser Regulierungsmöglichkeit liegt die Schwierigkeit in der Bestimmung des Parameters λ.

4.4.5 Selbstorganisierende Karten (SOM) / Kohonen-Netze

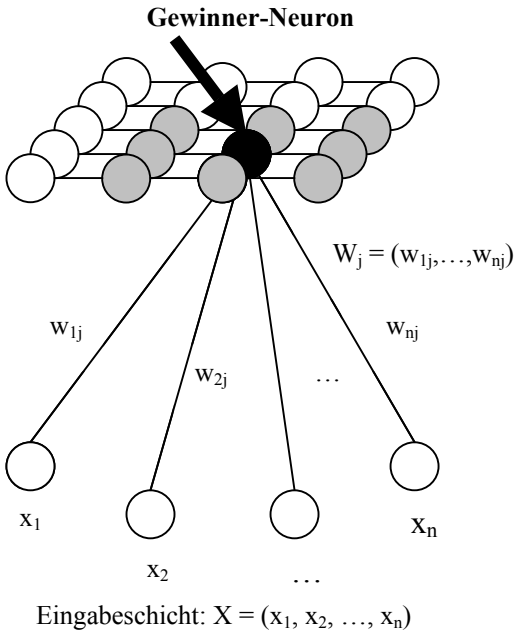
4.4.5.1 Prinzipien der selbstorganisierenden Karten

Die charakteristischen Merkmale einer SOM[129] sind die Nutzung der räumlichen Anordnung und damit der Nachbarschaftsbeziehungen der Neuronen. Außerdem wird im Gegensatz zu den bislang besprochenen KNN das unüberwachte Lernen angewendet. Ohne Vorgabe von gewünschten Ergebnissen für Trainingsdaten wird ein klassifizierendes Verhalten erlernt.
Kohonen-Netze bestehen aus zwei Schichten, einer Eingabeschicht und einer Kartenschicht. Die Kartenschicht liegt meistens, wie eben auch in Abbildung 4.15 zu sehen, in einer zweidimensionalen Darstellungsform[130] vor. Die Neuronen der Eingabeschicht sind mit den Neuronen der Kartenschicht voll vernetzt, d.h. es bestehen Verbindungen von jedem Eingabe- zu jedem Karten-Neuron.

---

[127] Vgl. SAS Institute Inc. (2000c), S. 181.
[128] Eine solche Fehlerfläche kann zu Oszillation oder Sprüngen auf der Fehlerfläche beim Lernen führen.
[129] Vgl. Kohonen (2001), S. 105 ff.
[130] Die Benutzung von quadratischen Karten ist am häufigsten anzutreffen, allerdings können nicht-quadratische Karten mitunter sogar zu besseren Ergebnissen führen.

Aufgrund der Vielzahl an Verbindungen wurde lediglich für ein Neuron die Darstellung der Verbindungen angedeutet:



Eine Netzstruktur wie in Abbildung 4.15 kann sehr gut zur Visualisierung und zum Auffinden von Ähnlichkeitsbeziehungen in einem hoch dimensionalen Eingaberaum benutzt werden. Dafür werden im Laufe des Trainingsprozesses die Daten in ungeordneter Reihenfolge dem Netz wiederholt präsentiert. Die Adaption eines Netzes erfolgt durch Herausbildung eines Erregungszentrums, dem sog. Gewinner-Neuron. Dieses Neuron besitzt die maximale, durch die Eingabe hervorgerufene Erregung.

Abb. 4.15: Architektur einer SOM.
Quelle: Vgl. Zell (2000), S. 180 und Lämmel, Cleve (2001), S. 228; eigene Darstellung.

Die Aufgabe des Trainingsprozesses ist somit die Bestimmung eines Gewinner-Neurons, das als Repräsentant einer Klasse von Daten fungiert.

4.4.5.2 Lernverfahren der selbstorganisierenden Karten

Zur Bestimmung des Gewinner-Neurons muss der Abstand zwischen dem Eingabemuster und den einzelnen Neuronen ermittelt werden. Zur Berechnung wird die Vektorschreibweise[131] verwendet. So wird das Eingabemuster, bestehend aus den Aktivierungen der Neuronen aus der Eingabeschicht, folgendermaßen dargestellt:

(4.39) $m_p = (m_{p1}, m_{p2}, \ldots, m_{pk}) = (o_1, o_2, \ldots, o_k)$.

Die Gewichte der Verbindungen von der Eingabeschicht zu dem Neuron j sind die Grundlage für den Vektor des Neurons j der Kartenschicht:

(4.40) $W_j = (w_{1j}, w_{2j}, \ldots, w_{kj})$.

Verfahren zur Bestimmung von Gewinner-Neuronen sind das maximale Skalarprodukt:

(4.41) $$\sum_{i=1}^{N} o_i w_{iz} = \max_j \sum_{i=1}^{N} o_i w_{ij}$$

[131] Bei den SOM besteht eine enge Verwandtschaft zu der lernenden Vektorquantifizierung.

oder das Minimum des euklidischen Abstands:

$$(4.42)\quad \sum_{i=1}^{N}\left(m_{pi}-w_{iz}\right)^2 = \min_j \sum_{i=1}^{N}\left(m_{pi}-w_{ij}\right)^2 .$$

Bei dem Verfahren des maximalen Skalarprodukts wird für jedes Neuron der Kartenschicht aus dem Vektor der Verbindungsgewichte und dem Vektor, der aus der Ausgabe der Eingabe-Neuronen resultiert, das Skalarprodukt gebildet und anschließend das Maximum bestimmt.

Der minimale euklidische Abstand ermittelt das Erregungszentrum durch die Minimierung des Gewichtsvektors der Verbindungen von den Eingabe-Neuronen zum Gewinner-Neuron abzüglich des Eingabevektors m_p, dargestellt in euklidischer Norm[132].

Nach der Bestimmung des Gewinner-Neurons werden im Trainingsprozess die Gewichte zur Eingabe und zu den benachbarten Neuronen modifiziert, da die räumliche Nähe auch den Grad der Gewichtsanpassung beeinflusst. Die Netzeingabe eines Neuron der Kohonen-Schicht setzt sich sowohl aus dem Einfluss der Eingabe als auch den Einflüssen der anderen Karten-Neuronen zusammen:

$$(4.43)\quad net_j = \sum_{i=1}^{N} o_i w_{ij} + \sum_{k=1}^{M} o_k w_{kj} + \theta_j .$$

Der Schwellenwert θ_j symbolisiert das Gewicht einer Verbindung eines aktivierten Neurons mit dem Neuron j. Für die Berechnung der Aktivierung wird die logistische Funktion benutzt, als Ausgabefunktion fungiert die Identität.

Die Modifizierung der Verbindungsgewichte bewirkt eine Verstärkung des Erregungszentrums, das nun deutlich sichtbarer als vor der Anpassung zu sehen ist. Die Verbindungsgewichte der weiter entfernt liegenden Neuronen werden nicht weiter verändert oder verringert.

4.5 Assoziations- und Sequenzanalyse

4.5.1 Einführung in die Assoziationsregeln

Um das Assoziationsproblem formal zu beschreiben, betrachtet man eine Datenmenge D von Transaktionen t. Jede Transaktion besteht aus einer Menge einzelner Elemente mit jeweils unterschiedlicher Häufigkeit des Auftretens.

Eine Assoziationsregel wird mit X ➔ Y bezeichnet, d.h. das Auftreten von Element X führt zu dem Auftreten des Elements Y. Element X befindet sich im Regelrumpf, ist also die Wenn-Bedingung. Element Y wird in den Regelkopf gesetzt und entspricht folglich der

[132] Vgl. Lämmel, Cleve (2001), S. 227.

Dann-Bedingung.[133] Damit eine Transaktion die Assoziationsregel X ➔ Y erfüllt, muss $(X \cup Y) \subseteq t$ gelten, es müssen also alle Elemente i.d.R. enthalten sein. Für die Bewertung einer Assozoationsregel existieren drei verschiedene Kriterien: Support, Konfidenz und Lift[134].

4.5.1.1 Support

Der Support[135] der Assoziationsregel ist derjenige Anteil aller Transaktionen, der die Regel erfüllt, d.h. die Anzahl der Transaktionen, in denen die Elemente der Regel vorkommen, dividiert durch die Gesamtheit der Transaktionen.

(4.44) $\textit{Support (X} \rightarrow \textit{Y)} = \frac{\left|\{t \in D \mid (X \cup Y) \subseteq t\}\right|}{|D|}$.

Häufig wird ein Mindestsupport gewählt, um Regeln auszublenden, die nur für einen kleinen Teil des Datenbestandes gelten.

4.5.1.2 Konfidenz

Die Konfidenz[136] der Assoziationsregel beschreibt, für welchen Anteil der Transaktionen, die X enthalten, die Assoziationsregel X ➔ Y gilt. Zur Berechnung wird die Anzahl der regelerfüllenden Transaktionen (Support (X ➔ Y)) durch die Anzahl aller Transaktionen, die X erhalten (Support (X)), dividiert.

(4.45) $\textit{Konfidenz (X} \rightarrow \textit{Y)} = \frac{\left|\{t \in D \mid (X \cup Y) \subseteq t\}\right|}{|t \in D \mid X \subseteq t|}$.

Analog zum Mindestsupport gibt es auch an dieser Stelle die Möglichkeit, eine Mindestkonfidenz zu bestimmen.

4.5.1.3 Lift

Der Lift[137] ist ein Maß für die Assoziation zwischen rechter und linker Seite. Er wird zur Vermeidung eines zufälligen Auftretens der Regel verwendet, also um die Unabhängigkeit der rechten Seite von der linken Seite auszuschließen. Die Werte des Lifts geben die Korrelation zwischen X und Y an. Allerdings ist eine hohe Korrelation nur ein Indikator für eine aussagekräftige Regel, da einem hohen Lift für gewöhnlich ein niedriger Support

[133] Vgl. Schinzer, Bange, Mertens (1999), S. 118.
[134] Im Anhang werden in Kapitel A.12 die Berechnungen des Supports, der Konfidenz und des Lifts durchgeführt.
[135] Vgl. Schinzer, Bange, Mertens (1999), S. 119.
[136] Vgl. Schinzer, Bange, Mertens (1999), S. 120.
[137] Vgl. SAS Institute Inc. (2002), S. 8-5.

gegenübersteht. Für die Bewertung einer Assoziationsregel sollten alle Maße berücksichtigt werden.

$$(4.46)\quad Lift\ (X \rightarrow Y) = \frac{\frac{\left|\{t \in D \mid (X \cup Y) \subseteq t\}\right|}{|t \in D \mid X \subseteq t|}}{\frac{|t \in D \mid Y \subseteq t|}{|D|}}.$$

4.5.2 Sequenzmuster

Ein Sequenzmuster ist eine gleichförmige Abfolge von Elementen in den Transaktionen verschiedener Kunden. Während die relevante Fragestellung für eine Assoziationsregel lautet, „welche Artikel werden zusammen gekauft?“, wird bei Sequenzmustern gefragt, „welche Artikel werden nacheinander gekauft?“.[138] Zur Beantwortung dieser Frage und für die Entdeckung von Sequenzmustern muss eine Transaktionsdatenbank neben den Elementen einer Transaktion auch Zusammengehörigkeitsmerkmale (Transaktionszeit und Kundennummer) aufweisen. Die Speicherung der Elemente der Transaktion erfolgt mit binären Variablen[139], da für Sequenzmuster quantitative Erfassungen der Elemente unbedeutend sind. Lediglich der zeitliche Rahmen ist für die Analyse ausschlaggebend. Die nach Transaktionszeit sortierten Transaktionen eines Kunden werden als Sequenz bezeichnet.

Analog zum Auffinden von Assoziationsregeln lässt sich bei Sequenzmustern das Bewertungskriterium des Mindestsupports angeben, um Muster mit nur geringem Gültigkeitsbereich innerhalb des Datenbestandes auszublenden.

Möglichkeiten zur Verbesserung der Aussagekraft lassen sich durch eine Neudefinition der Zeitstruktur[140] erreichen.

[138] Vgl. Schinzer, Bange, Mertens (1999), S. 121.
[139] Ein typisches Beispiel könnte demnach lauten: 1 = gekauft, 0 = nicht gekauft.
[140] Eine Neudefinition von Tages- zu Wochenabständen würde z.B. den Vorteil bringen, die wöchentlichen Einkäufe zu erfassen.

5. Modellbewertung

5.1 Bewertung der Klassifizierungsleistung

Zur Bewertung der Modellgüte stehen verschiedene statistische Kriterien zur Verfügung:

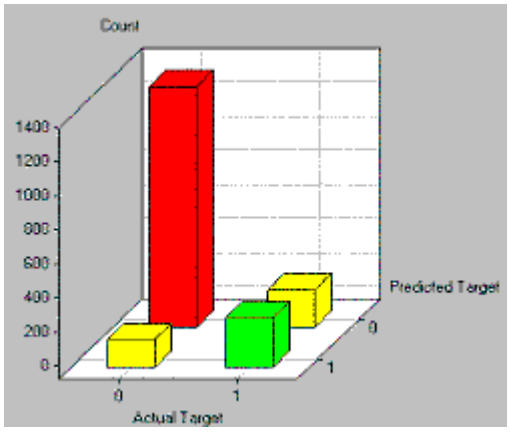
Root ASE	Valid:Root ASE	Test:Root ASE	Schwarz Bayesian Criterion	Misclassification Rate	Valid:Misclassification Rate	Test:Misclassification Rate
0.2969364291	0.3071402691	0.2949419802		0.1098993289	0.1140939597	0.1073825503

Abb. 5.1: Assessment-Kriterien.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Die Misclassification Rate, im Beispiel von Abbildung 5.1 gerundete 0.11, also 11 Prozent falsch klassifizierte Fälle, wird in der später erörterten Confusion-Matrix berechnet.

Außerdem stehen die Kriterien Quadratwurzel der durchschnittlichen Fehlerquadrate (Root ASE)[141] und das Schwarz Bayes'sche Kriterium (SBC)[142] zur Verfügung.

Der Erfolg eines Modells lässt sich auch anhand der Klassifizierungsleistung messen. Die Confusion-Matrix lässt sich aufgrund ihrer leicht zugänglichen Darstellungsform gut interpretieren.



Confusion-Matrix

Tatsächlicher Wert \ Prognostizierter Wert	0	1
0	η_{TN}	η_{FP}
1	η_{FN}	η_{TP}

TN = True Negative
TP = True Positive
FN = False Negative
FP = False Positive

Abb. 5.2: Confusion-Matrix.
Quelle: Screenshot SAS® Enterprise Miner™; SAS Institute Inc. (2000a), S. 75; eigene Darstellung.

So steht η_{TN} für den Anteil an Fällen, die negativ, dargestellt durch den Wert 0, klassifiziert worden sind und diese Ausprägung auch tatsächlich angenommen haben. η_{TP} steht somit entsprechend für die korrekt klassifizierten positiven Ausprägungen, durch den Wert 1 dargestellt. Fälle bei denen ein Fehler erster Art vorliegt, d.h. ein tatsächlich positiver Wert wurde negativ klassifiziert, werden durch η_{FN} dargestellt.

η_{FP} steht für die Fälle, denen fälschlicherweise ein positiver Wert zugewiesen wurde und somit einen Fehler der zweiter Art darstellen.

---

[141] $Root\ ASE = \sqrt{ASE} = \sqrt{\frac{SSE}{n}}$, mit SSE als Sum of Squared Error.

[142] $SBC = n \cdot \ln\left(\frac{SSE}{n}\right) + k \cdot \ln(n)$, mit k als Modell-Freiheitsgrade.

Klassifikationsprobleme[143] werden mit Wahrscheinlichkeiten zwischen Null und Eins bewertet. Es muss von Seiten des Anwenders eine Schranke, ein sog. Cut-Off, bestimmt werden, der angibt, ab wann ein prognostizierter Wert als Eins interpretiert wird. Ein Maß zur Bewertung der Vorhersagegenauigkeit ist die ROC-Kurve[144]. Mit Hilfe der Sensitivität[145] und der Spezifikation[146] lässt sich die Klassifikationsgüte grafisch darstellen[147].

5.2 Draw Lift Charts

Für ein beliebiges Klassifikationsproblem stehen verschiedene Draw Lift Charts zur Verfügung: %Response, %Captured Response und Lift Value.

Der %Response-Chart gibt für jedes Perzentil die Wahrscheinlichkeit für eine Klassifizierung des Zielereignisses an. Im Falle der besten zehn Prozent, d.h. der eindeutigsten Fälle, lässt sich mit einer Wahrscheinlichkeit von 84 Prozent (vgl. Abb. 5.3) eine Zuordnung für das Erfolgsereignis angeben. Dagegen gibt die Baseline lediglich den durchschnittlich zu erwartenden Wert an, ca. 48 Prozent. Der Chart der %Captured Response ist wie eine Lorenz-Kurve zu interpretieren. Will man z.B. 80 Prozent der potentiell erfolgreichen Ereignisse erfassen, reichen dafür die besten 60 Prozent der Fälle aus (vgl. Abb. 5.4).

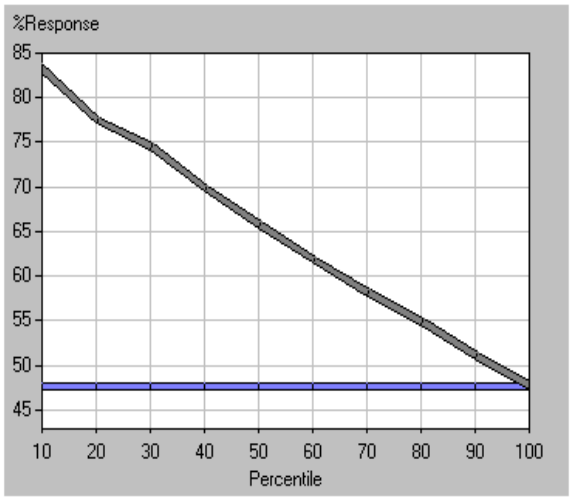


Abb.5.3: %Response.
Quelle: Screenshot SAS® Enterprise Miner™.

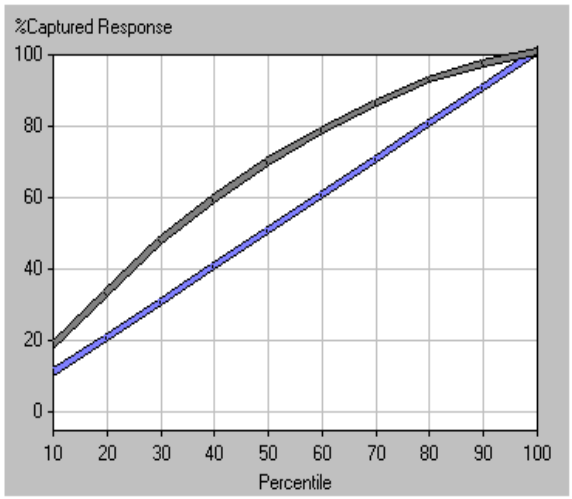


Abb. 5.4: %Captured Response.
Quelle: Screenshot SAS® Enterprise Miner™.

[143] Logistische Modelle bzw. binäre Zielvariablen.
[144] Receiver Operating Characteristic.
[145] Genauigkeitsmaß für vorhergesagte „ True Positive-Werte“ in Abhängigkeit zu der Gesamtheit an aktuell positiv prognostizierten Werten.
[146] Genauigkeitsmaß für vorhergesagte „True Negative-Werte“ in Abhängigkeit zu der Gesamtheit an aktuell negativ prognostizierten Werten.
[147] Im Anhang wird in Kapitel A.13 eine ROC-Kurce gezeigt, außerdem werden weitere Erläuterungen angefügt.

Der Lift-Wert gibt die Verbesserung des Modells gegenüber der Baseline für die jeweiligen Perzentile an. Im ersten Perzentil performt das Modell in Abbildung 5.3 ungefähr 1,75-mal so gut wie die Baseline.

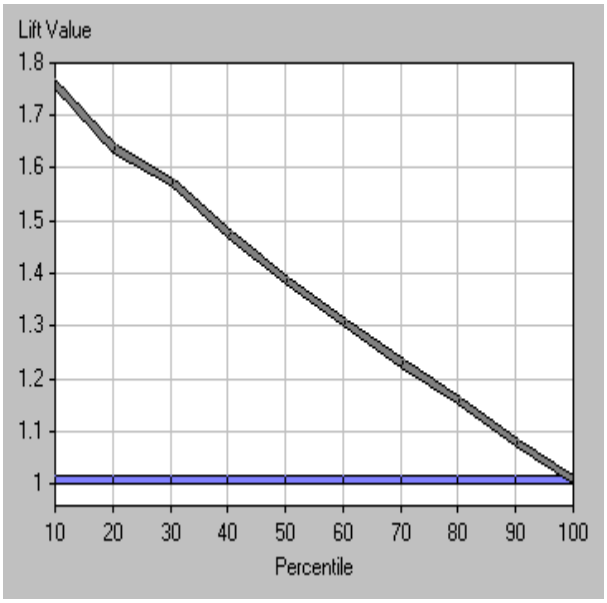


Abb. 5.5: Lift Value.
Quelle: Screenshot SAS® Enterprise Miner™.

Wurde eine Profit-Matrix definiert, lassen sich darüber hinaus noch der zu erwartende Gewinn und der Return on Investment (ROI) ablesen.
Alle Ergebnisse lassen sich in kumulierter und nicht-kumulierter Form angeben.

6. Fallstudien

Im Rahmen dieser Arbeit werden drei Fallstudien besprochen. Zuerst wird mit der Optimierung einer Mailing-Aktion ein typisches marketingspezifisches Problem, das mit den Methoden des Data Minings gelöst werden kann, dargestellt. Die dafür notwendigen Schritte „Pre-Processing“, „Modellierung“ und „Modellbewertung“ kommen ebenso zum Tragen wie das Anwenden von neuen Daten im Scoring-Prozeß. Die beiden nachfolgenden Fallstudien, die sich mit den Besonderheiten der künstlichen neuronalen Netze und dem Entscheidungsbaumverfahren beschäftigen, werden entsprechend dem Umfang in Kapitel 4 besonders ausführlich präsentiert. Sie werden nicht als betriebwirtschaftliche Studien beschrieben, sondern zeigen einen Ausschnitt der bestehenden Möglichkeiten, die der SAS® Enterprise Miner™ bereitstellt. So werden im Falle der neuronalen Netze unterschiedliche Netzarchitekturen, Trainingsverfahren und das Early Stopping untersucht, währenddessen für das Entscheidungsbaumverfahren Bäume mit unterschiedlichen Attributauswahlverfahren generiert und mögliche Vorteile eines Bagging-Prozesses gegenüber Einzelmodellen evaluiert werden. Die zur Lösung des jeweiligen Problems verwendeten Diagramme zeigen stets nur eine Lösungsalternative, über andere Optionen, Einstellungen und Methoden lassen sich die Ergebnisse sicherlich noch optimieren. Die Bearbeitung der Fallstudien lief lediglich unter der Absicht, Lösungsmöglichkeiten aufzuzeigen. Eine ausführlichere Beschreibung der Fallstudien wird im Anhang ab Kapitel A.23 vorgenommen, dort werden ebenfalls die in den Beispielen vorkommenden Variablen beschrieben.

6.1 Fallstudie A: Optimierung einer Mailing-Aktion

Szenario:

Eine Versandhandelfirma verkauft via monatlich erscheinendem Katalog Möbel und Haushaltswaren. Nun soll eine neue Kampagne gestartet werden, die dem Motto „Schönes Abendessen“ gewidmet ist. Unter diesen Begriff fällt Küchenware wie Geschirr und Besteck. Es würden zu hohe Kosten anfallen, wenn jeder registrierte Kunde diesen extra erscheinenden Katalog erhalten würde, deshalb entschließt sich das Management, den Kundenstamm zu selektieren. Es sollen nur diejenigen berücksichtigt werden, bei denen eine hinreichend hohe Kaufwahrscheinlichkeit gegeben ist.

In der Datenbank des Unternehmens sind die Kundenkäufe mit zweijährigem Zeithorizont aufgezeichnet, des Weiteren sind Daten zur familiären Situation inklusive Einkommensangaben enthalten. In dem verwendeten Datensatz sind die Variablen „Kitchen Product“,

„Dishes Purchase“ und „Flatware Purchase“ vorhanden, die angeben, wie häufig in den letzten zwei Jahren Produkte verkauft wurden, die zu dem neuen Katalog passen. Diese Variablen werden zu einer neuen binären Variable transformiert, die als Zielvariable angeben soll, ob ein Kauf von Artikeln aus dem neuen Katalog zu erwarten ist.

Anhand dieses Problems lässt sich zeigen, dass der Erfolg der Data Mining-Aktion quantifizierbar nachzuweisen ist. Es werden nicht nur Angaben über das verbesserte Ergebnis bei der Antworthäufigkeit der Kunden gemacht, sondern darüber hinaus noch aufgezeigt, welcher Gewinn aus der Data Mining-Analyse zu erwarten ist. Ausgehend von der Annahme, dass die Konzeption und der Druck des Katalogs sowie die Porto-Gebühr 10,00 $ Kosten pro Katalog verursachen würden und im Falle eines Kaufes durchschnittlich 90,00 $ erwirtschaftet werden, ergeben sich folgende drei Fälle:

Kunde wird bei Mailing-Aktion nicht berücksichtigt:		**Kunde wird bei Mailing-Aktion berücksichtigt, aber kein Kauf:**		**Kunde wird bei Mailing-Aktion berücksichtigt, danach Kauf:**	
Kosten:	0,00 $	Kosten:	10,00 $	Kosten:	10,00 $
Umsatz:	0,00 $	Umsatz:	0,00 $	Umsatz:	90,00 $
Nettoergebnis:		**Nettoergebnis:**		**Nettoergebnis:**	
0,00 $		**-10,00 $**		**80,00 $**	

Abb. 6.1: Ausgangssituation zur Berechnung einer Profit-Matrix.
Quelle: Eigene Darstellung.

Diese Profit Matrix wird in die Analyse einbezogen, so dass der Gewinn oder der resultierende Return on Investment (ROI) nun zu bestimmen sind.

Eine Response-Optimierung krankt unter den in Abschnitt 3.1.2 beschriebenen, seltenen Zielereignissen. Daher wird ein Datensatz verwendet, bei dem die Merkmalsausprägung 1 (Kauf) gegenüber dem Nicht-Kauf mit der Ausprägung 0 im Verhältnis 54 Prozent zu 46 Prozent überrepräsentiert ist. Der dadurch resultierende Bias wird bereinigt, indem ein Prior-Vektor eingesetzt wird, der dann das korrekte Verhältnis (12 Prozent zu 88 Prozent) widerspiegelt. Der Prior-Vektor nutzt für die Umrechnung den Offset-Faktor.

Zur Lösung des beschriebenen Problems kann ein Data Mining-Prozess mit folgendem Diagramm dienen:

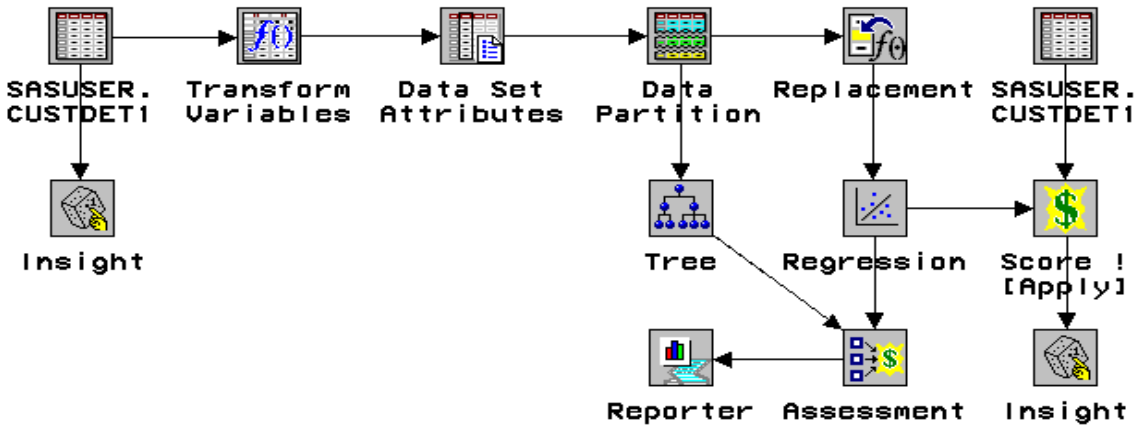


Abb. 6.2: SAS® Enterprise Miner™-Diagramm zur Selektion der Kunden aufgrund ihrer Kaufwahrscheinlichkeiten.
Quelle: Screenshot SAS® Enterprise Miner™.

Unter der Annahme, dass das Mailing nur für die besten zehn Prozent der Kunden durchgeführt werden soll, wird gemäß Abbildung 6.2 die Regressionsanalyse für den Scoring- Prozess verwendet. Die Regressionsanalyse erreicht sowohl in der %Response als auch bei dem Profit bessere Ergebnisse. Es werden im Vergleich zu dem Entscheidungsbaumverfahren höhere Antwortwahrscheinlichkeiten bzw. Gewinne erzielt (vgl. Abb. 6.3).

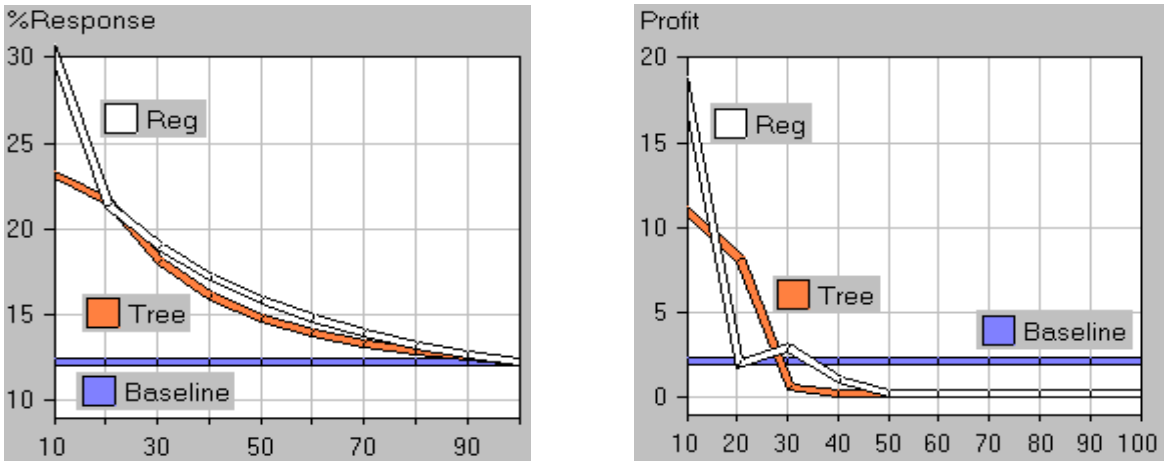


Abb. 6.3: %Response- und Profit-Chart für Regressionsanalyse und Entscheidungsbaumverfahren.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Anhand der Profit-Matrix ist nun ablesbar, dass durch das Data Mining in den besten zehn Prozent mit einem durchschnittlichen Gewinn von 18 $ zu rechnen ist. Sollten die besten 20 Prozent ausgewählt werden, bringt der Entscheidungsbaum mit einem Gewinn von 11 $ im ersten Percentil, weitere 8 $ im zweiten und insgesamt durchschnittlich 9,50 $ Gewinn das bessere Resultat. Im Output Fenster des SAS®-Systems lassen sich die Klassifizierungen ablesen.

Abbildung 6.4 zeigt exemplarisch die besten 25 Beobachtungen.

Obs	DINEBIN	P_DINEBIN1	P_DINEBIN0
1	1	1.00000	0.00000
2	1	0.99932	0.00068
3	1	0.99839	0.00161
4	1	0.99419	0.00581
5	1	0.99280	0.00720
6	1	0.98637	0.01363
7	1	0.98637	0.01363
8	1	0.98315	0.01685
9	1	0.97920	0.02080
10	1	0.94142	0.05858
11	1	0.92837	0.07163
12	1	0.87181	0.12819
13	1	0.87181	0.12819
14	0	0.87181	0.12819
15	1	0.87181	0.12819
16	1	0.84581	0.15419
17	1	0.84581	0.15419
18	1	0.84581	0.15419
19	1	0.84581	0.15419
20	1	0.81565	0.18435
21	1	0.81565	0.18435
22	1	0.81565	0.18435
23	1	0.81565	0.18435
24	1	0.81565	0.18435
25	0	0.81565	0.18435

Abb. 6.4: Beobachtungen mit der höchsten Kaufwahrscheinlichkeit.
Quelle: Screenshot SAS® System.

6.2 Fallstudie B: Funktionsweise der KNN

Aufgrund der Komplexität der neuronalen Netze werden nun einige der ab Abschnitt 4.4 beschriebenen Charakteristika, nämlich Netzwerkarchitektur, Lernregel und Regulierbarkeit, wieder aufgegriffen und modelliert. Dafür stehen Bankdaten zur Verfügung, die 2000 Einträge über ihre Kunden mit Angaben über die Anzahl der monatlichen Transaktionen, das monatliche Einkommen, Einträge über Investment-Beteiligungen oder den Saldo des Sparkontos enthalten. Dazu kommt eine weitere Variable, die als binäre Target-Variable fungiert und angibt, ob ein Interesse an einem neuen Investment-Produkt besteht. Die Hälfte der Kunden zeigt sich interessiert an neuen Produkten, hat also die Ausprägung 1. Die Bank nimmt allerdings an, dass lediglich zwölf Prozent der Kunden an einem solchen Produkt Interesse zeigen, und lediglich für die Modellanpassung wird eine solche Aufteilung[148] gewählt.

6.2.1 Auswahl der Netzwerkarchitektur bei NRBF-Netzen

Normalisierte Radiale-Basisfunktionen-Netze können gemäß Abschnitt 4.4.2.2 bezüglich der Höhe und der Breite mittels eines Parameters und der Gewichte modifiziert werden. Dementsprechend lassen sich für die Gauss'schen Funktionen und damit auch für die NRBF-Architektur fünf verschiedene Bedingungen formulieren: gleiche Breite und gleiche Höhe, gleiche Höhe, gleiche Breite, gleiches Volumen und freier Verlauf.

[148] Der dadurch resultierende Bias wird durch einen Prior-Vektor bereinigt.

In Abbildung 6.5 werden diese Modifikationsmöglichkeiten dargestellt. Das Prozess-Diagramm zeigt lediglich das Auswahlverfahren bei der Modellanpassung. Die zu scorenden Daten könnten im Vorgehen analog zu Fallstudie A bewertet werden.

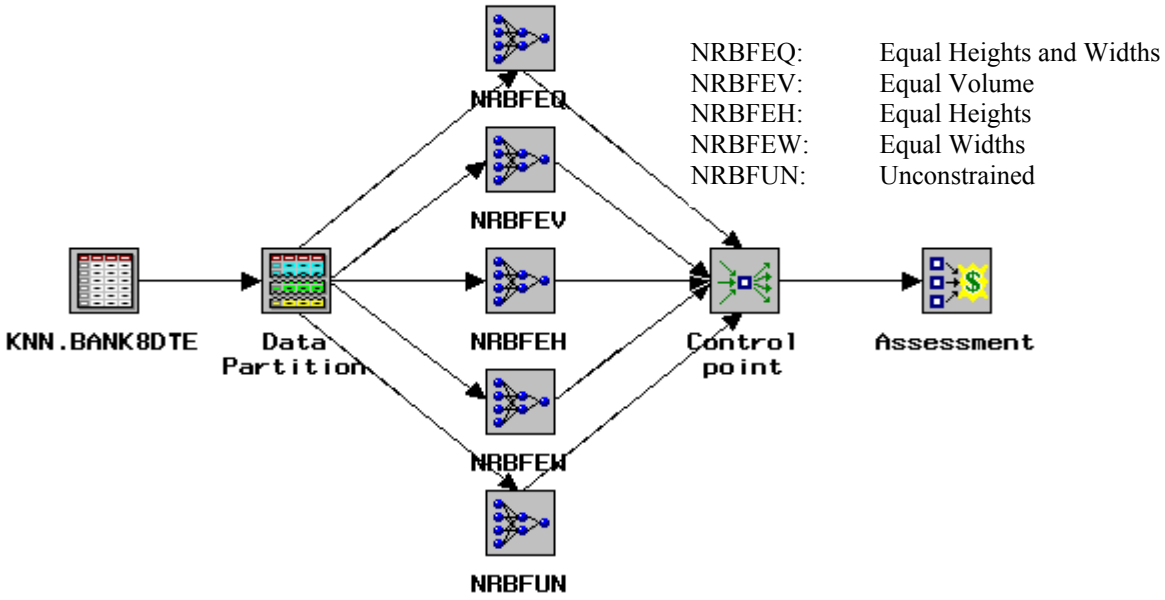


Abb. 6.5: NRBF-Netzwerkarchitektur.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Der Draw Lift Chart zeigt, dass für das in Abbildung 6.6 gezeigte Prozess-Diagramm die NRBF-Netzwerkarchitektur mit der Bedingung, das die Gauss'schen Funktionen die gleiche Breite besitzen, das größte Erfolgspotential für das erste Percentil besitzen.

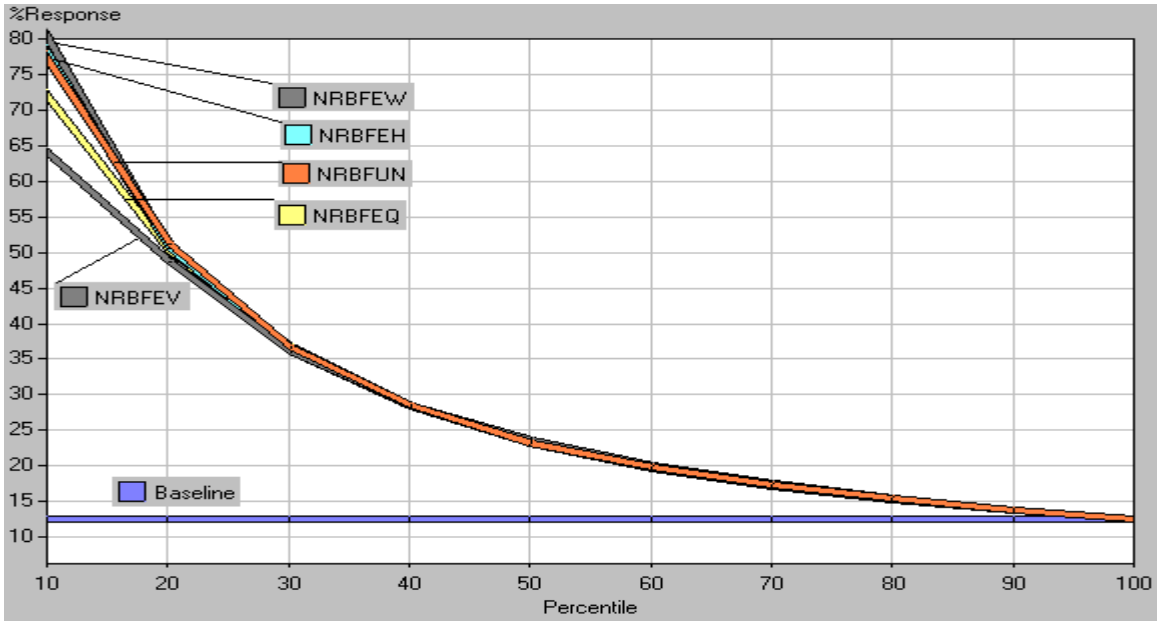


Abb. 6.6: Draw Lift Charts für die verschienen Normalisierten Radialen-Basisfunktionen-Netze.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Die Verbesserung ist im Vergleich zur Baseline (80 Prozent zu 12 Prozent) beträchtlich. Aber auch zwischen den verschiedenen NRBF-Netzen existieren erhebliche Unterschiede.

So schneidet das Netzwerk mit der Bedingung „gleiches Volumen“ mit 15 Prozentpunkten schlechter ab. Für die Modellbewertung lassen sich die in Abbildung 6.7 gezeigten statistischen Auswertungen heranziehen. Neben den in Abschnitt 5.1 vorgestellten Kriterien (Root ASE, Misclassification Rate und Schwarz Bayesian Criterion) stehen nun weitere Fehlerberechnungen, die als Indikator für die Modellgüte dienen, sowie weitere Informationen wie Anzahl der Freiheitsgrade, Frequenzsumme oder Akaike’s Information Criterion (AIC) zur Verfügung.

		NRBFEQ			NRBFEV			NRBFEH			NRBFEW			NRBFUN		
	Fit Statistic	Training	Validation	Test	Training	Validatio	Test	Training	Validation	Test	Training	Validation	Test	Training	Validatio	Test
1	[TARGET=ACQUIRE]	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	Average Profit	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
3	Misclassification Rate	0.26	0.27667	0.2233	0.23	0.22667	0.1933	0.215	0.20333	0.1867	0.2025	0.19667	0.1667	0.1625	0.17333	0.1733
4	Average Error	0.329	0.41396	0.377	0.35602	0.65946	0.5409	0.33602	0.58949	0.5201	0.33693	0.4639	0.4326	0.31956	0.40782	0.3804
5	Average Squared Error	0.10155	0.11425	0.1118	0.10966	0.13338	0.1258	0.10345	0.11544	0.1119	0.1042	0.11368	0.1138	0.09674	0.11582	0.1102
6	Sum of Squared Errors	81.2404	68.5526	67.058	87.7292	80.0272	75.494	82.7579	69.2655	67.164	83.3637	68.2108	68.294	77.3895	69.4901	66.099
7	Root Average Squared Error	0.31867	0.33802	0.3343	0.33115	0.36521	0.3547	0.32163	0.33977	0.3346	0.32281	0.33717	0.3374	0.31103	0.34032	0.3319
8	Root Final Prediction Error	0.3444	.	.	0.3597	.	.	0.34936	.	.	0.35152	.	.	0.3404	.	.
9	Root Mean Squared Error	0.33179	0.33802	0.3343	0.34572	0.36521	0.3547	0.33578	0.33977	0.3346	0.33747	0.33717	0.3374	0.32604	0.34032	0.3319
10	Error Function	263.202	248.376	226.22	284.813	395.678	324.56	268.817	353.696	312.07	269.541	278.342	259.56	255.65	244.691	228.27
11	Mean Squared Error	0.11008	0.11425	0.1118	0.11952	0.13338	0.1258	0.11275	0.11544	0.1119	0.11388	0.11368	0.1138	0.1063	0.11582	0.1102
12	Maximum Absolute Error	0.97587	0.99878	0.9957	0.98562	1	1	0.99076	1	1	0.99209	1	1	0.95788	0.99954	0.9995
13	Final Prediction Error	0.11861	.	.	0.12938	.	.	0.12205	.	.	0.12357	.	.	0.11587	.	.
14	Divisor for ASE	800	600	600	800	600	600	800	600	600	800	600	600	800	600	600
15	Model Degrees of Freedom	31	.	.	33	.	.	33	.	.	34	.	.	36	.	.
16	Degrees of Freedom for Error	369	.	.	367	.	.	367	.	.	366	.	.	364	.	.
17	Total Degrees of Freedom	400	.	.	400	.	.	400	.	.	400	.	.	400	.	.
18	Sum of Frequencies	400	300	300	400	300	300	400	300	300	400	300	300	400	300	300
19	Sum Case Weights * Frequencies	800	600	600	800	600	600	800	600	600	800	600	600	800	600	600
20	Akaike's Information Criterion	325.202	.	.	350.813	.	.	334.817	.	.	337.541	.	.	327.65	.	.
21	Schwarz's Baysian Criterion	448.938	.	.	482.531	.	.	466.535	.	.	473.251	.	.	471.343	.	.

Abb. 6.7: Statistische Auswertungen für die verschiedenen NRBF-Architekturen.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Diese Auswertung ist im KDD-Ansatz, der stets einen iterativen Prozess verlangt, natürlich lediglich ein erster Schritt bei der Modellanpassung. Es sollten mehrere Anpassungen folgen oder das Beispiel mit einer anderen Stichprobe wiederholt werden.

6.2.2 Auswahl des Lernverfahrens bei MLP-Netzwerkarchitekturen

Für die Auswahl des Lernverfahrens wird eine Multilayer Perceptron-Architektur verwendet. Diese besitzt zwei verdeckte Schichten mit jeweils zwei Neuronen. Außerdem ist die Input- mit der Output-Schicht direkt verbunden, d.h. das Netzwerk besitzt Shortcut

Connections. Dieses MLP wird nun mit sieben verschiedenen Lernverfahren trainiert: Konjugierter Gradientenabstieg, Backpropagation, sowie dessen Modifikationen RPROP und Quickprop, Levenberg-Marquard und die beiden Newton-Verfahren Quasi-Newton und Newton-Raphson.

Folgendes Diagramm kann zur Lösung des Problems herangezogen werden:

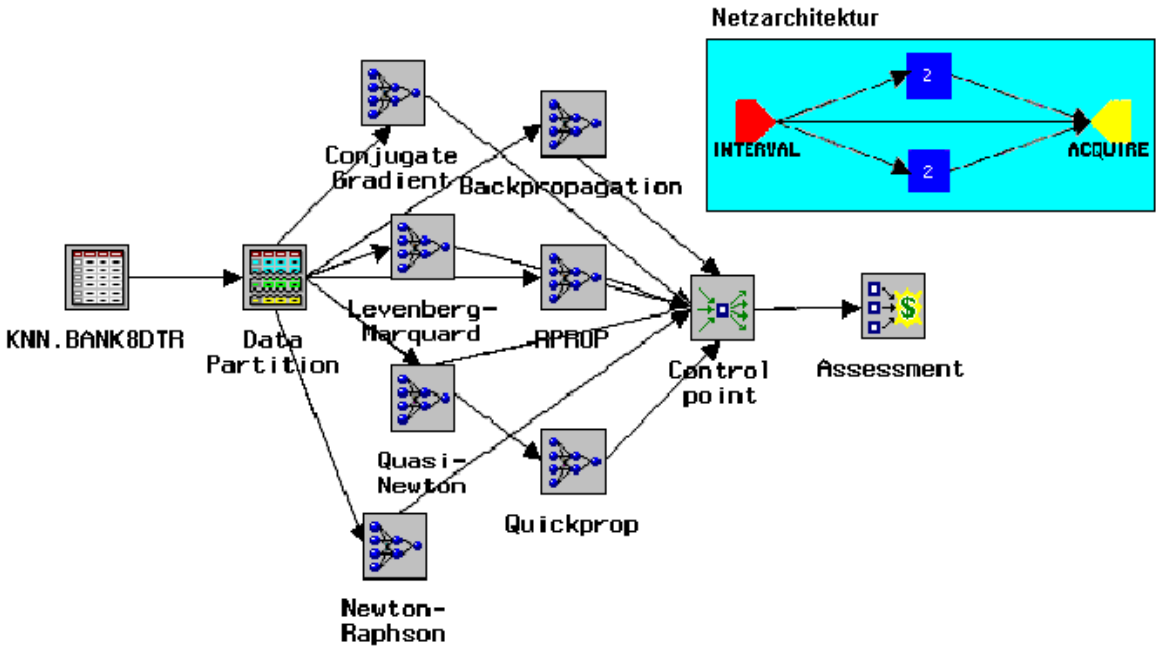


Abb. 6.8: SAS® Enterprise Miner™-Diagramm für die Bestimmung des Lernverfahrens sowie die Netzarchitektur der einzelnen KNN.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Im Assessment-Knoten wird eine statistische Zusammenfassung mit den Kriterien Root ASE, Schwarz Bayesian Criterion und Misclassification Rate (vgl. Abschnitt 5.1) angeboten. Die Auswahl eines geeigneten Kriteriums zur Bewertung der Modellgüte ist nicht eindeutig zu klären, da dies auch immer von der Fragestellung abhängig ist. Abbildung 6.9 verdeutlicht die Schwierigkeiten bei der Bewertung und Auswahl der verschiedenen Kriterien: Die Rangfolge der besten Werte ist bei jedem der drei Kriterien Root ASE, SBC und Misclassification Rate unterschiedlich.

Description	Target	Root ASE	Schwarz Bayesian Criterion	Misclassification Rate
Conjugate Gradient	ACQUIRE	0.3282494693	882.75276943	0.23
Backpropagation	ACQUIRE	0.3566243646	1010.4255712	0.29875
Levenberg-Marquard	ACQUIRE	0.3198504052	867.31752709	0.22625
Quasi-Newton	ACQUIRE	0.3271109515	883.84437996	0.22625
Newton-Raphson	ACQUIRE	0.3267191851	884.3133941	0.22
RPROP	ACQUIRE	0.3275105999	887.96209776	0.22875
Quickprop	ACQUIRE	0.3309336876	898.25397688	0.22875

Abb. 6.9: Bewertung der Lernverfahren: Konjugierter Gradientenabstieg, Backpropagation, Levenberg-Marquard, Quasi-Newton, Newton-Raphson, RPROP und Quickprop.
Quelle: Screenshot SAS® Enterprise Miner™.

Die Modellbewertung sollte deshalb auf Basis verschiedener Kriterien getroffen werden. Vergleicht man die sieben Lernverfahren der MLP-Architekturen[149] auf der Basis von Draw Lift Charts, so erhält man:

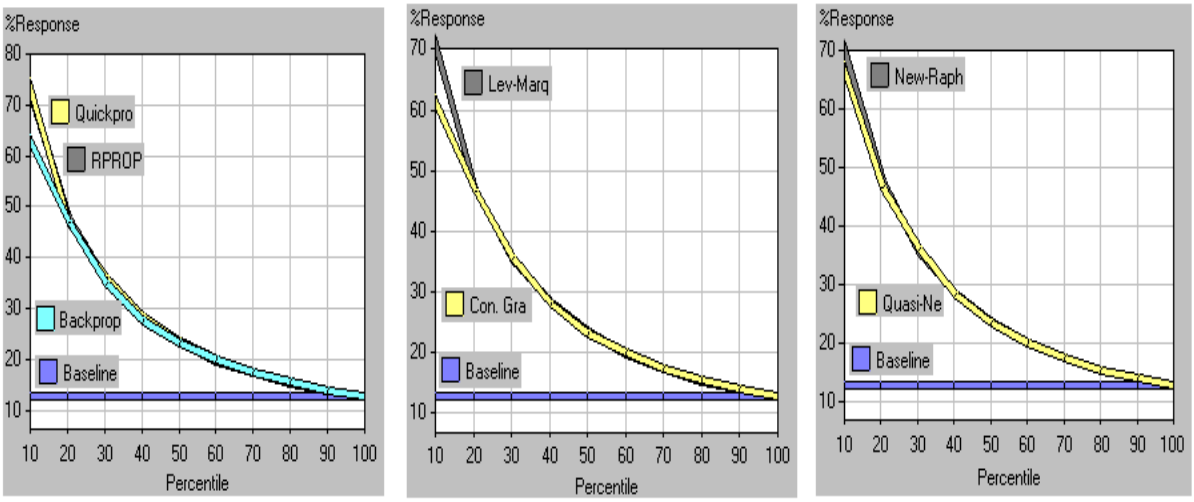


Abb. 6.10: Modellbewertung mit Draw Lift Charts.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Die unterschiedlichen Lernverfahren der MLP-Netzarchitekturen lassen unterschiedlich gut entwickelte Modelle entstehen. Vergleicht man diese Ergebnisse mit den Resultaten der NRBF-Netze, erkennt man, dass sich mit den NRBF-Netzen – in diesem Beispiel – eine bessere Modellanpassung erreichen lässt. Als Nachteil erweist sich dort allerdings die größere Streuung in den Ergebnissen. Selbstverständlich ließen sich weitere Modelle entwickeln. So wurden beispielsweise die NRBF-Netze immer mit der Default-Trainingstechnik entwickelt. Die große Anzahl an verschiedenen Aktivierungs- und Kombinationsfunktionen, verschiedene Trainingstechniken, Regulierung mit Weight Decay und nicht zuletzt die Anzahl der verdeckten Schichten und Neuronen bieten ein riesiges Spektrum an Möglichkeiten, die bei der Modellanpassung bei KNN denkbar sind.
An dieser Stelle muss abermals auf die Notwendigkeit einer iterativen Modellanpassung hingewiesen werden. Allerdings sollte auch der Faktor Zeit nicht aus dem Auge verloren werden. Die Datenbeschaffung und -bereinigung erweist sich in der Realität als sehr zeitaufwendig, so dass die eigentliche Modellierung oftmals nur sehr beschränkt zeitlich möglich ist. Daher empfiehlt es sich, Rücksprachen mit dem fachlich Verantwortlichen über das erwünschte Ergebnis zu halten.
Die letztgenannte Variationsmöglichkeit bei der Entwicklung von neuronalen Netzen, die Anzahl der Neuronen, wird den Abschluss in der Fallstudie über KNN bilden.

[149] Aus Gründen der Übersichtlichkeit wurden Bachpropagation und seine Modifikationen zusammengefasst, ebenso die verschiedenen Newton-Verfahren sowie Levenberg-Marquard und der Gradientenabstieg.

6.2.3 Early Stopping

Um ein deutlich hervortretendes Early Stopping zu erlangen, muss ein ausreichend komplexes Modell ein Overfitting bewirken. Die Daten erhalten folgende Aufteilung: 90 Prozent Training, 10 Prozent Validierung. Zur Bestimmung einer Overfitting erzeugenden Modell-komplexität kann die Faustregel „*kein Overfitting bei mindestens 10 Fällen für jeden zu schätzenden Parameter*“ herangezogen werden. Gemäß Abschnitt 4.4.2.1 hat ein h-Neuronen-Netzwerk mit einer verdeckten Schicht h (k + 2) + 1 Parameter[150]. Daraus folgt die Gleichung $h(k+2)+1=\frac{n}{10}$ mit n als Grundgesamtheit aller Fälle und für die Bestimmung der Neuronen-Anzahl: $h=\frac{n-10}{10(k+2)}$. Allerdings wird bei binären Zielvariablen nicht die Grundgesamtheit, sondern die Anzahl der am seltensten vorkommenden Ausprägung[151] gewählt, also: $h=\frac{\min(n_1,n_0)-10}{10(k+2)}$. Bei gewählten 90 Prozent Trainingsdaten ergibt sich für h der Wert 8,9. Ab 9 Neuronen wird das Modell ein Overfitting aufweisen. Außerdem wird ein KNN mit Default-Einstellungen als Vergleich herangezogen. Auf diese Weise soll gezeigt werden, dass bei einem zu komplexen Modell der Effekt des Auswendiglernens auftritt. In Abbildung 6.11 ist das dafür benötigte SAS® Enterprise Miner™-Diagramm samt Netzarchitektur abgebildet:

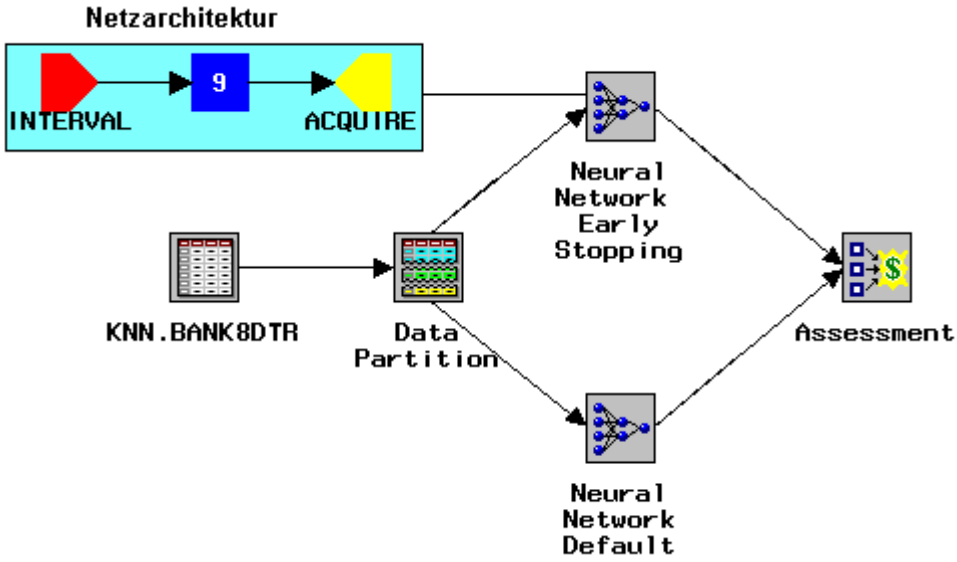


Abb. 6.11: SAS® Enterprise Miner™-Diagramm für ein KNN mit einer verdeckten Schicht mit 9 Neuronen und ein Default-KNN.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

[150] Die Anzahl der Input-Variablen k ist 8.
[151] In dem verwendeten Beispiel sind die Event- und Non-Event-Fälle gleich verteilt.

Abbildung 6.12 zeigt den erwünschten Effekt. Ab ca. 18 Iterationsschritten setzt das Early-Stopping ein, d.h. der Validierungsfehler beginnt zu steigen, bei gleichzeitigem weiterem Absinken des Trainingsfehlers. Dagegen verlaufen die Trainings- und Validierungsfehler bei einem neuronalen Netz mit Default-Einstellungen weitgehend parallel, der Effekt des Auswendiglernens tritt nicht ein.

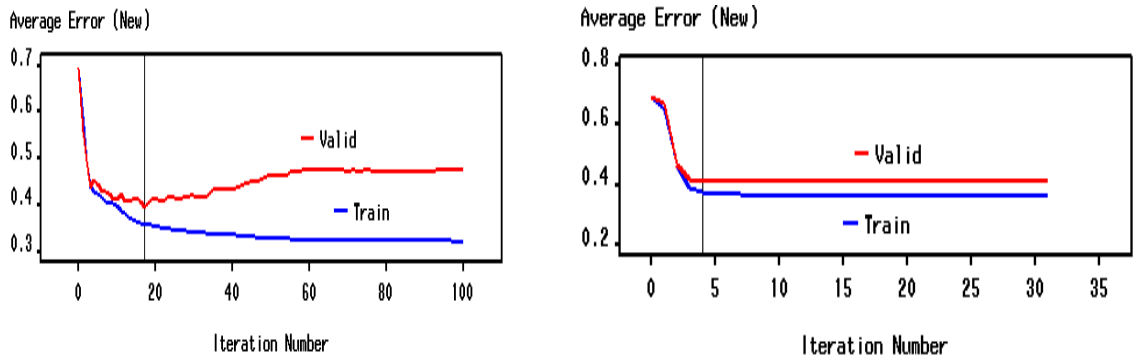


Abb. 6.12: Early Stopping-KNN vs. Default-KNN.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

6.3 Fallstudie C: Entscheidungsbaumverfahren

In der Fallstudie über Entscheidungsbäume werden die Bestimmung des Auswahlmaßes und das Bagging mit Bäumen demonstriert. Die dafür verwendeten Daten[152] stammen aus einer Befragung, die die Gewohnheiten im Umgang mit dem Internet untersucht. Als Target-Variable können entweder Aussagen über Cookie-Präferenzen oder die tatsächlich hier verwendete Variable MAJORDER benutzt werden. Diese Variable gibt an, ob der Befragte einen Kauf über 100 $ über das Internet abgewickelt hat. Die Daten enthalten diverse Aussagen über die Nutzungsgewohnheiten, die technische Ausstattung und verschiedenste persönliche Angaben.

6.3.1 Bestimmung des Auswahlmaßes

Aus den vorhandenen Daten sollen drei unterschiedliche binäre Bäume mit den zur Verfügung stehenden Attributauswahlmaßen χ^2, Entropie und Gini-Index modelliert werden.

In dieser Fallstudie soll neben der Interpretation von Entscheidungsbäumen und Entscheidungsregeln gezeigt werden, inwieweit die Wahl des Attributauswahlmaßes das Baumwachstum beeinflusst.

[152] Die Daten stammen aus der 10th GVU (Graphic Visualization and Usability Center) WWW User Survey. [Copyright 1994-1998 Georgia Tech Research Corporation. All rights reserved.]
Vgl. SAS Institute Inc. (2000a), S. 14 ff.

Dazu kann das folgende Prozess-Diagramm herangezogen werden:

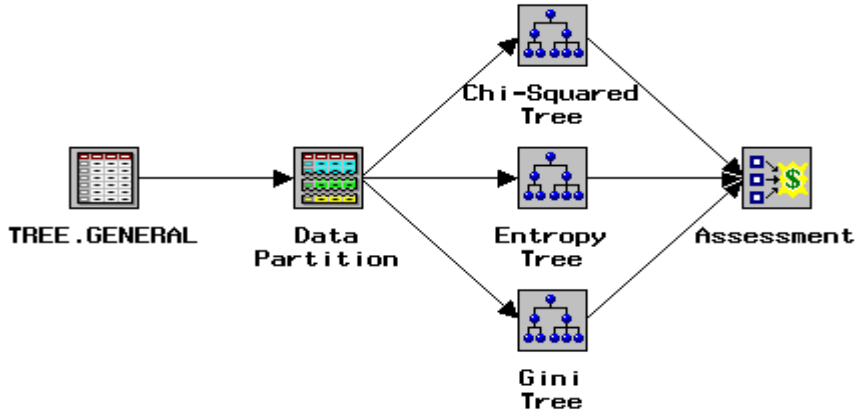


Abb. 6.13: SAS® Enterprise Miner™-Diagramm für die Bestimmung des Auswahlmaßes.
Quelle: Screenshot SAS® Enterprise Miner™.

Der SAS® Enterprise Miner™ liefert für den Entscheidungsbaum mit dem χ^2-Auswahlmaß die in Abbildung 6.14 gezeigte Baumstruktur.

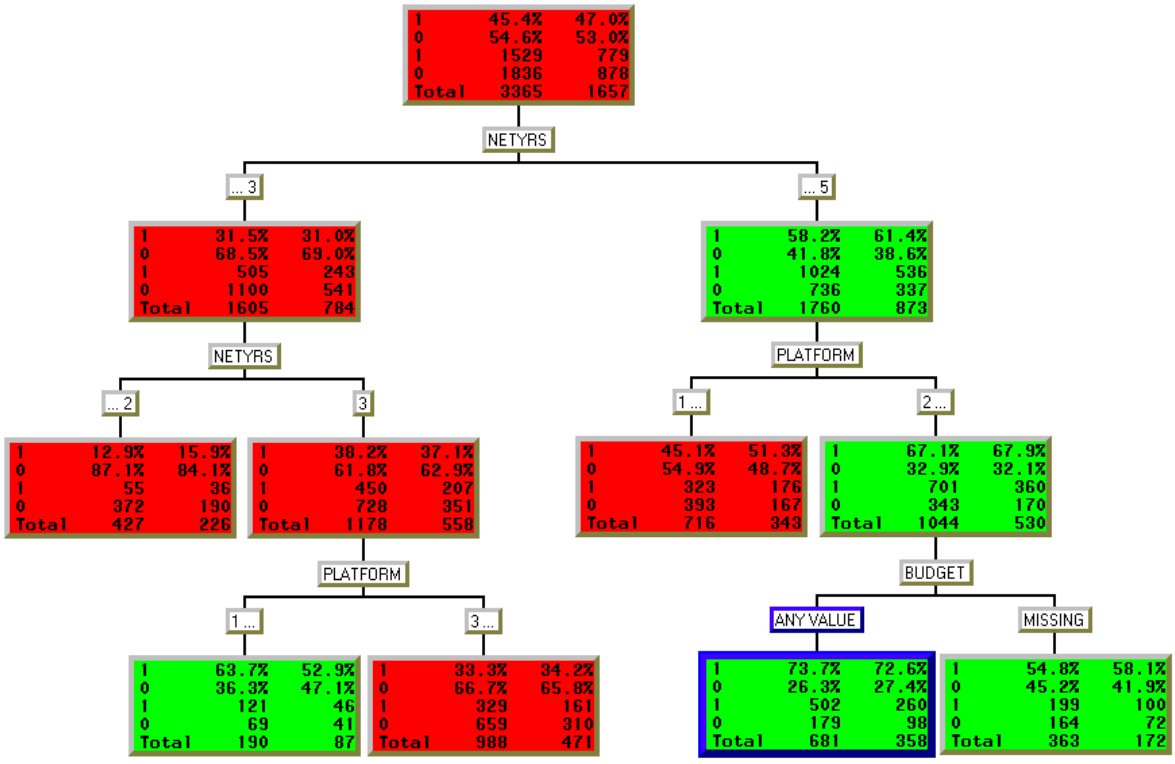


Abb. 6.14: Entscheidungsbaum mit χ^2-Auswahlmaß.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

Dieser binäre Baum hat sechs Blätter – drei mit Prognose 1 und drei mit Prognose 0. Die Anzahl der Jahre im Netz ist das Kriterium, das die beste Unterscheidung liefert und deshalb an der Wurzel des Entscheidungsbaumes steht. Anhand des Baumes lässt sich zeigen, dass einzelne Variablen auf dem Weg von der Wurzel zu einem Blatt mehrfach als

Trennkriterium eingesetzt werden können[153]. Außerdem können nach einer Trennung auf dem restlichen Weg unterschiedliche Variablen verwendet werden[154].

Das dick umrandete Blatt (BUDGET ➔ [ANY VALUE]) enthält die Entscheidungsregel mit der besten Klassifikation. 502 von 681 Befragten (73,7 Prozent) werden mit 1, also Internetkauf über 100 $, klassifiziert. Die dazu passende Entscheidungsregel lautet:

IF *BUTGET* is ANY VALUE AND *PLATFORM* is one of 2, 4, 5, 6, 8, 10, 14 AND *NETYRS* is one of 4, 5 THEN: 1 = 72,7 %; 0 = 26,3 %.

Übersetzt bedeutet diese Entscheidungsregel folgendes:

Wenn ein Budget existiert und als Plattform entweder Mac, OS2, Unix, PC Unix, NT, oder Win98 verwendet werden und die Anzahl der Jahre, die der User im Internet ist, vier bis sechs oder größer gleich sieben ist, dann ist die Wahrscheinlichkeit gleich 72,7 Prozent, dass ein Kauf über 100 $ erfolgt.

Das Auswahlmaß, das die beste Modellanpassung ermöglicht, lässt sich wie in der Fallstudie über neuronale Netze mit dem Assessment-Knoten bestimmen. Natürlich lassen sich auch bei den Entscheidungsbäumen eine Vielzahl von weiteren Möglichkeiten wie Stoppkriterien, Einsatz von Surrogat-Splits oder Verwendung von N-Way-Bäumen zur Modellanpassung vornehmen.

Interessant ist bei Bäumen mit unterschiedlichen Attributauswahlmaßen das dadurch entstehende Wachstum der Blätter und Äste.

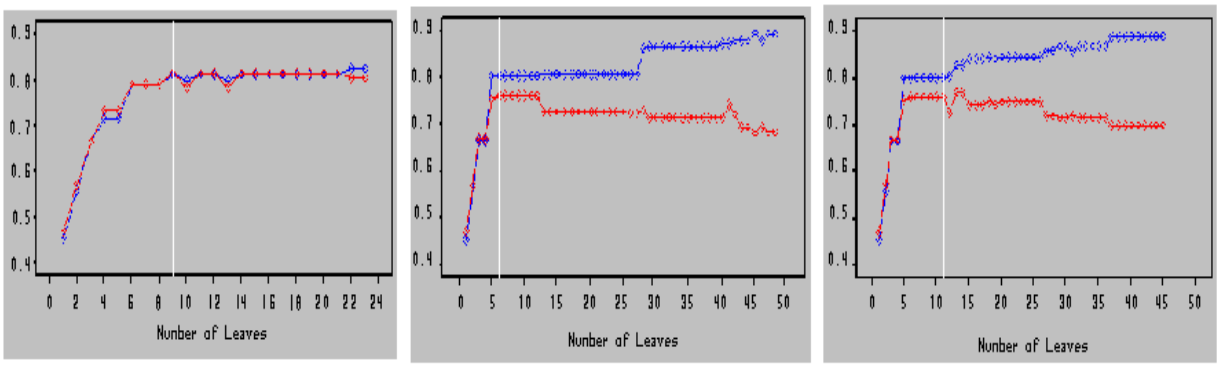


Abb. 6.15: Training vs. Validierung von Entscheidungsbäumen mit den Auswahlmaßen χ^2, Entropy und Gini-Index.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

So wachsen Bäume durch den Gini-Index oder Entropy sehr viel stärker als durch das χ^2-Maß. Allerdings sind die Modelle mit vielen Blättern nicht sehr aussagekräftig, so dass sie mittels Pruning gestutzt werden. Die besten Baummodelle werden mit einer Blattanzahl von 7 (Entropy) bis 12 (Gini-Index) generiert.

[153] Die Variable NETYRS wird beispielsweise in dem linken Ast doppelt verwendet.
[154] NETYRS ➔ NETYRS ➔ PLATFORM im linken Ast und NETYRS ➔ PLATFORM ➔ BUDGET im rechten Ast.

6.3.2 Bagging

Bagging-Modelle können bessere Ergebnisse liefern als Einzelmodellen. Dieser Ansatz soll nun durch ein Bagging-Verfahren mit einem Entscheidungsbaum verdeutlicht werden, der gegen einen einzelnen Baum antritt.

Dafür wird ein Diagramm wie in Abbildung 6.16 benötigt:

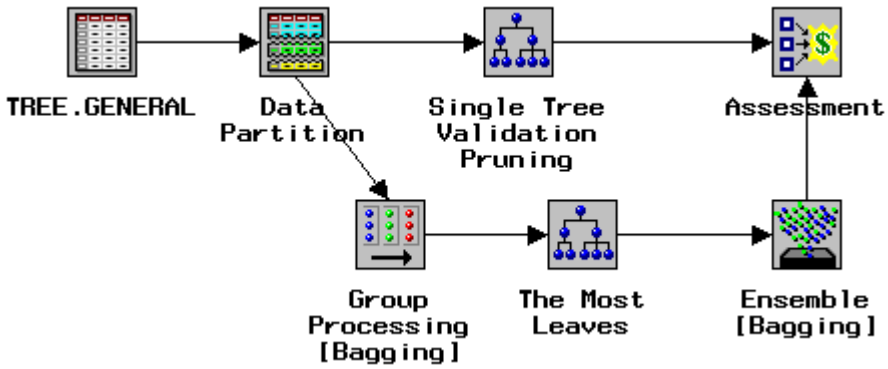


Abb. 6.16: SAS® Enterprise Miner™-Diagramm zur Durchführung eines Bagging-Prozesses.
Quelle: Screenshot SAS® Enterprise Miner™.

Das Bagging wird mit 15 Durchläufen durchgeführt. Wie in Abschnitt 4.3.4 beschrieben, wird bei dem dafür verwendeten Entscheidungsbaum auf das Pruning verzichtet. Der Ansatz, dass durch die wiederholte Durchführung von Entscheidungsbäumen und anschließender Durchschnittsbildung ein besseres Ergebnis erzielt werden kann als durch ein Einzelmodell, zeigt Abbildung 6.17. Der Draw Lift Chart %Response zeigt ein um 20 Prozentpunkte besseres Ergebnis für das Bagging-Modell im Vergleich zum Einzelmodell.

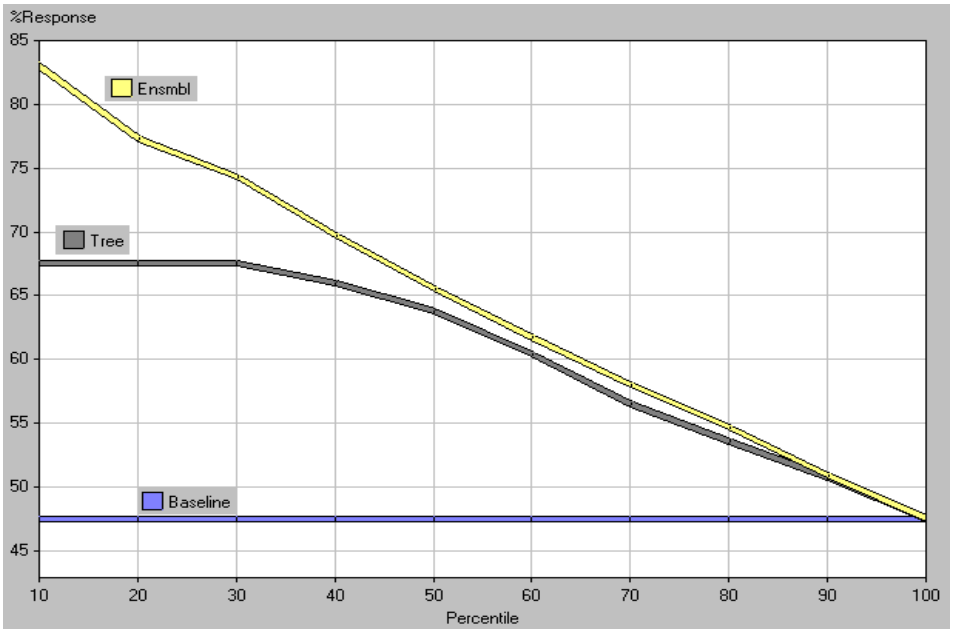


Abb. 6.17: Bagging-Modell vs. Einzelmodell.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

7. Zusammenfassung

Data Mining bietet im operativen und strategischen Entscheidungsprozess die Möglichkeit, aus großen Datenbeständen Muster, Informationen oder Wissen zu generieren. Ergebnisse aus Vorgängen, die aufgrund der Datenbeschaffenheit, d.h. Daten mit massiver Größe, sonst im Verborgenen bleiben würden, können somit gefunden werden.

Der Fokus dieser Arbeit liegt auf Prognose- und Klassifikationsmodellen, da diese meines Erachtens, sowohl in der theoretischen Auseinandersetzung als auch in der praktischen Umsetzung, besonders mächtig bezüglich ihrer zur Verfügung stehenden Möglichkeiten sind. Darüber hinaus wurden die anderen Aufgabenstellungen des KDD vorgestellt. Der einzuhaltende Umfang erlaubte es aber nicht, näher, beispielsweise mit Fallstudien, auf die Assoziationsanalyse oder die Clusterung einzugehen. Diese Verfahren sind aber gleichberechtigter Teil des Data Mining-Prozesses und kommen in der praktischen Anwendung, z. B. bei Warenkorbanalysen oder im Bereich des Marketings bei Markt-, Kunden- und Preissegmentierungen, häufig zum Tragen.

Meine Intention bei der Darstellung des Data Mining-Ansatzes ist, einen Einblick in die Verfahren und ihre Methoden zu bieten, um so den Umfang und die Tiefe der Einsatzmöglichkeiten darzustellen, die sicherlich in einigen Bereichen erst am Anfang ihrer Entwicklung stehen.

Stets wurde auf die Notwendigkeit hingewiesen, die Modellanpassung als iterativen Prozess zu verstehen. Dies gilt nicht nur für die Modellierung selbst, sondern auch für die Gesamtheit des KDD-Ansatzes. Während einer Data Mining-Analyse sollte immer die Bereitschaft vorhanden sein, das Pre-Processing oder die Modellierung entsprechend der Modellbewertung zu verändern.

In der Modellbewertung und der möglichen Bestimmung der Rentabilität liegt meinem Ermessen nach ein weiterer wichtiger Vorteil des Data Minings, gerade im Vergleich zu den in der Einleitung beschriebenen betriebswirtschaftlich geprägten Analyseverfahren, wo diese Möglichkeit, wenn überhaupt, nur beschränkt möglich ist.

An dieser Stelle sollte noch einmal auf die Data Mining-Definition von Fayyad aus Abschnitt 2.1 eingegangen werden, die besagt, dass die erzielten Ergebnisse den Forderungen nach Validität, Neuartigkeit, Nützlichkeit und Verständlichkeit genügen sollen. Diese Ansprüche an die Ergebnisse sollten dann auch abschließend in die Bewertung mit einfließen. So ließe sich die Validität beispielsweise in die Kategorien Akkuratheit, Robustheit und statistische Signifikanz weiter unterteilen und dies zusätzlich in der Modellbewertung berücksichtigen. Die genannten Ansätze sollten an dieser Stelle

kurz erwähnt werden, da deren Zusammenwirken bei der Exploration von Wissen hilfreich sein kann.

Trotz der weitreichenden und mächtigen Methoden, die einer Data Mining-Analyse zur Verfügung stehen, sind diesem Ansatz aber auch deutliche Grenzen gesetzt. Neben technischen Problemen, meistens in Bezug auf die Datenqualität, wurden Hoffnungen, die an eine Aktienkursprognose oder Vorhersagen in chaotischen Systemen geknüpft wurden, nur sehr beschränkt erfüllt. Dies sollte allerdings nicht als grundsätzliche Schwäche des Data Minings ausgelegt werden.

A N H A N G

A.1 Erstellung von Vorhersage- bzw. Klassifikationsmodellen VII

A.2 Bias und Varianz eines Schätzers VIII

A.3 Integration von Data Warehouse, Data Mining und OLAP X

A.4 OLAP XI

4.1 OLAP-Würfel XI

4.2 Das Slice-Verfahren XI

A.5 Einsatzgebiete des Data Mining XII

A.6 Cross Validation XIII

A.7 Complete Case Analysis XIV

A.8 Biologisches Neuron XV

A.9 Aktivierungsfunktionen XVI

9.1 Logistische Funktion XVI

9.2 Tangens Hyperbolicus XVI

A.10 Topologien XVII

A.11 Herleitung der Backpropagation-Regel XVIII

A.12 Support, Konfidenz und Lift einer Assoziationsregel XX

A.13 ROC-Kurve XXI

A.14 Die SEMMA-Methode XXII

A.15 Eine kurze Erläuterung des SAS® Enterprise Miner™ XXIII

A.16 Die Knoten der Sample-Gruppe XXIV

16.1 Input Data Source XXIV

16.2 Sampling und Data Partition XXIV

A.17 Die Knoten der Explore-Gruppe XXV

17.1 Distribution Explorer, Multiplot und Insight XXV

17.2 Association XXV

17.3 Variable Selection XXV

17.4 Link Analysis XXV

A.18 Die Knoten der Modify-Gruppe XXVI
18.1 Data Set Attributes XXVI
18.2 Transform Variables und Filter Outlier XXVI
18.3 Clustering und SOM / Kohonen XXVI
18.4 Time Series XXVI

A.19 Die Knoten der Model-Gruppe XXVII
19.1 Regression, Tree und Neural Network XXVII
19.2 Princomp / Dmneural XXVII
19.3 User-Defined Model XXVII
19.4 Ensemble XXVIII
19.5 Memory-Based Reasoning XXVIII
19.6 Two Stage Model XXVIII

A.20 Die Knoten der Assess-Gruppe XXIX
20.1 Assessment XXIX
20.2 Reporter XXIX

A.21 Die Knoten der Score-Gruppe XXX
21.1 Score XXX
21.2 C*Score XXX

A.22 Die Knoten der Utility-Gruppe XXXI
22.1 SAS Code XXXI
22.2 Control Point und Subdiagram XXXI
22.3 Group Processing XXXI
22.4 Data Mining Database XXXI

A.23 Prozessbeschreibungen der Fallstudien - Fallstudie A XXXII

A.24 Prozessbeschreibungen der Fallstudien - Fallstudie B XXXVIII
24.1 Auswahl der Netzwerkarchitektur bei NRBF-Netzen XXXVIII
24.2 Auswahl des Lernverfahren bei MLP-Netzwerkarchitekturen XL
24.3 Early Stopping XLII

A.25 Prozessbeschreibungen der Fallstudien Fallstudie C: XLIV
25.1 Bestimmung des Attributauswahlmaßes XLIV
25.2 Bagging XLVI

A.1 Erstellung von Vorhersage- bzw. Klassifikationsmodellen

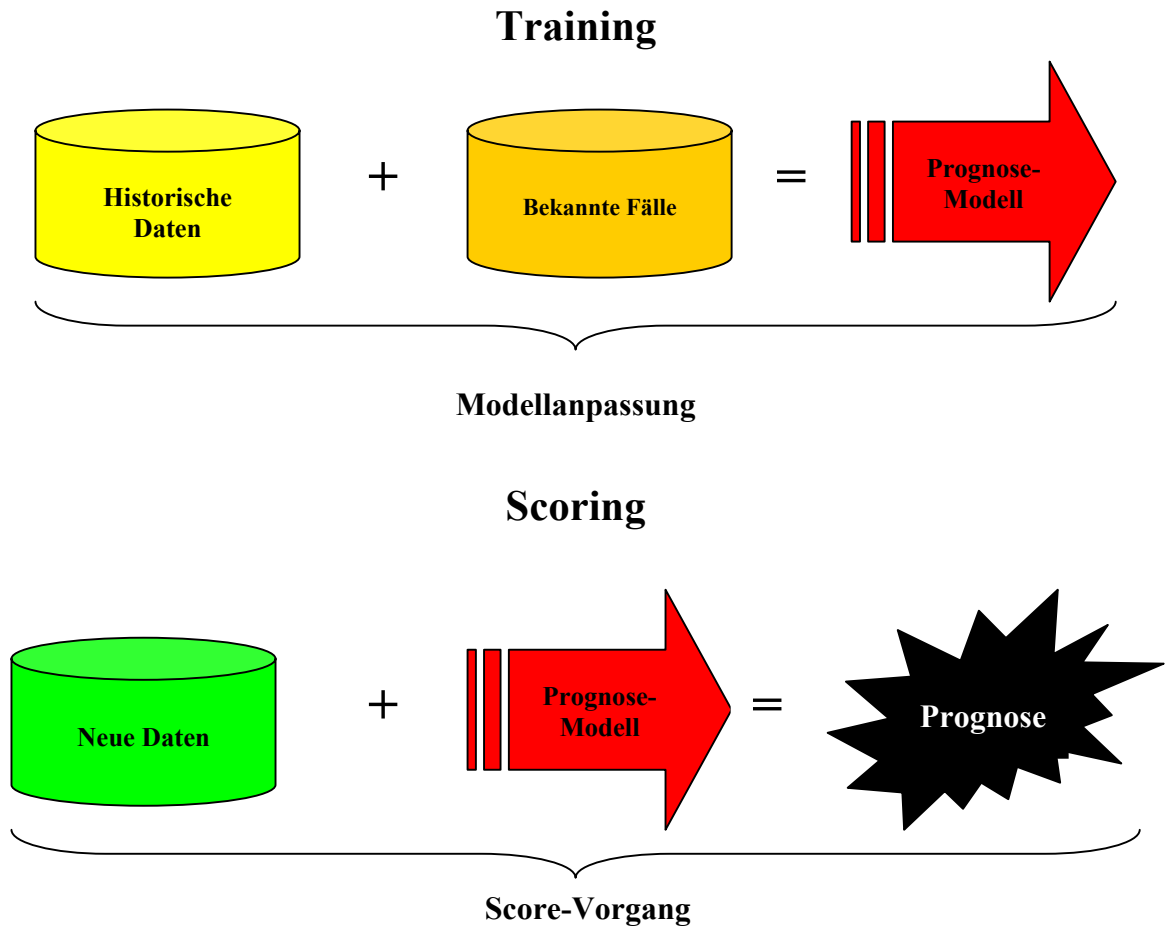


Abb. A.1: Ablauf eines Vorhersage- bzw. Klassifikationsmodells.
Quelle: Eigene Darstellung.

Vorhersage- bzw. Klassifikationsmodelle entstehen während des Trainingsprozesses. Im Sinne des überwachten Lernens werden dafür historische Daten, bei denen die Auswirkungen durch bekannte Fälle vorliegen, verwendet. Das Prognosemodell, beispielsweise ein KNN, eine Regressionsanalyse oder ein Entscheidungsbaum, wird entwickelt, indem die Regressoren dem Regressand oder den abhängigen Variablen angepasst werden.

Existiert nach einer Modellanpassung ein Prognosemodell, kann dieses für das Scoren von neuen Fällen verwendet werden, um auf diese Weise eine Vorhersage oder eine Klassifikation zu erhalten.

A.2 Bias und Varianz eines Schätzers

Die Bestimmung der Parameter θ einer Funktion f erfolgt bei parametrischen Verfahren oftmals mit dem Zielkriterium des geschätzten Mean Squared Error (MSE)[1] einer Regression:

(A.1) $$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - f(X_t,\hat{\theta}))^2 .$$

Der durchschnittliche Fehler (MSE) jeder Modelschätzung lässt sich in die Komponenten unsystematischer Fehler, MSE_u, und systematischer Fehler, MSE_s, zerlegen:

(A.2) $$\begin{aligned}\text{MSE} &= E[(y - f(X,\hat{\theta}))^2] \\ &= E[(y - F(X) + F(X) - f(X,\hat{\theta}))^2] \\ &= E[(y - F(X))^2] + E[(F(X) - f(X,\hat{\theta}))^2] + \underbrace{2E[(y - F(X))(F(X) - f(X,\hat{\theta}))]}_{0} \\ &= \text{MSE}_u + \text{MSE}_s.\end{aligned}$$

Der erwartete unsystematische Fehler MSE_u wird durch einen Störterm erzeugt und ist aufgrund seiner zufälligen Entstehung durch kein Modell approximierbar. Die Regressionsgüte wird daher lediglich durch den systematischen Fehler MSE_s gemessen. Dieser Fehler lässt sich weiter unterteilen in den Bias und die Varianz eines Schätzers[2]. Für einen Parameterschätzer $\hat{\theta}$ gilt:

(A.3) $$\text{MSE}_s[\hat{\theta}] = \text{Bias}[\hat{\theta}]^2 + \text{Var}[\hat{\theta}].$$

Der Bias eines Schätzers ist als die Abweichung des Erwartungswertes des Schätzers von dem tatsächlichen Wert definiert. Die Varianz des Schätzers ist die durchschnittliche quadrierte Abweichung des Schätzers von seinem Erwartungswert. Somit berechnet sich der Bias eines Punktschätzers $\hat{\theta}$ aus

(A.4) $$\text{Bias}[\hat{\theta}] = \text{E}[\hat{\theta}] - \theta$$

und für seine Varianz gilt:

(A.5) $$\text{Var}[\hat{\theta}] = \frac{1}{T}(\hat{\theta} - E[\hat{\theta}])^2 .$$

Der Funktionenschätzer f lässt sich analog zu dem Parameterschätzer in einen Bias und eine Varianz zerlegen:

(A.6) $$\text{MSE}_s[f] = E[(F(X) - f(X,\hat{\theta}))^2] = \text{Bias}[f(X,\hat{\theta})]^2 + \text{Var}[f(X,\hat{\theta})].$$

---

[1] Der MSE bestimmt sich aus der Minimierung der durchschnittlichen quadratischen Abstände zwischen den tatsächlichen Werten der Variable y und den durch die Funktion $f(\text{X},\hat{\theta})$ geschätzten Werten $\hat{y}$.

[2] Vorraussetzung für eine Unterteilung ist die Interpretation der Funktion f als einen Funktionenschätzer für die Funktion F.

Der Bias von *f* gibt dabei an, wie weit der Funktionsverlauf von *f* im erwarteten Mittel von dem Funktionsverlauf von F entfernt ist und stellt somit die systematische Verzerrung der Approximation dar. Die Varianz von *f* gibt den Bereich an, den der Funktionsverlauf von *f* durch die Approximation der wahren Funktion F annehmen kann.

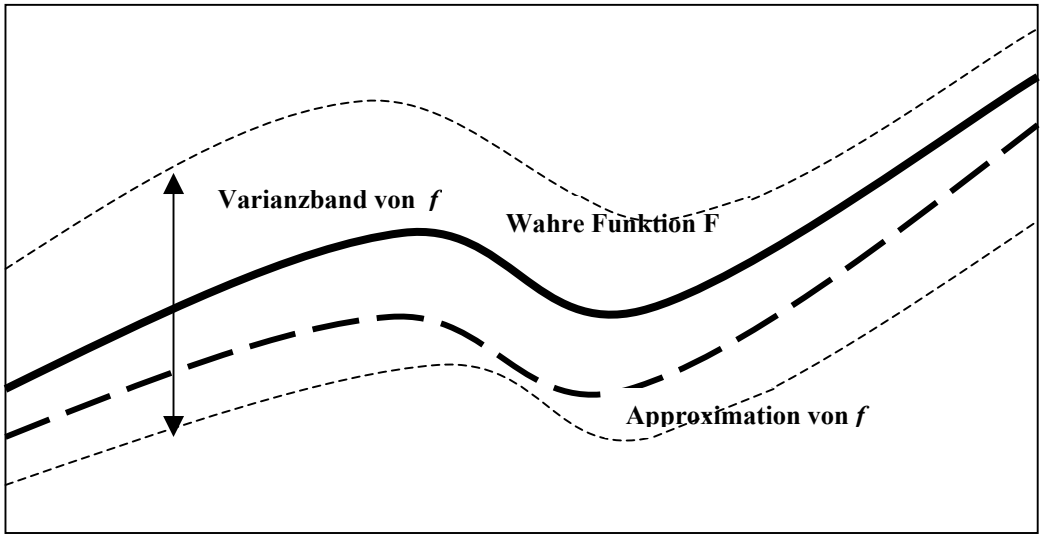


Abb. A.2: Bias und Standardabweichung eines Funktionsschätzers *f*.
Quelle: Anders (1995), S. 7; eigene Darstellung.

Alle innerhalb des Varianzbandes liegenden Funktionsverläufe sind aufgrund der gegebenen Daten möglich.

A.3 Integration von Data Warehouse, Data Mining und OLAP

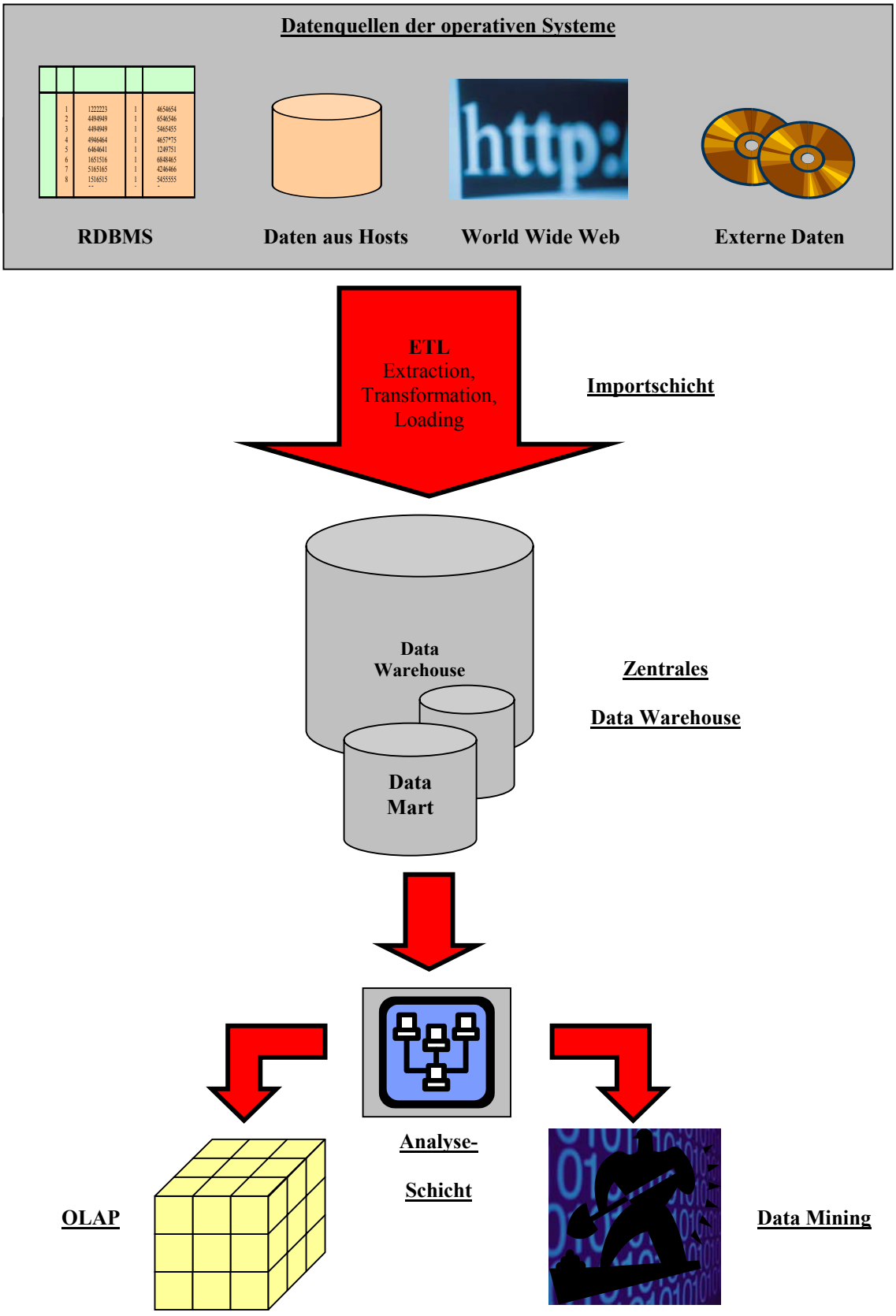


Abb. A.3.: Architektur von Data Warehouse, Data Mining und OLAP.
Quelle: Schütte, Rotthowe, Holten (2001), S. 8; eigene Darstellung.

A.4 OLAP

4.1 OLAP-Würfel

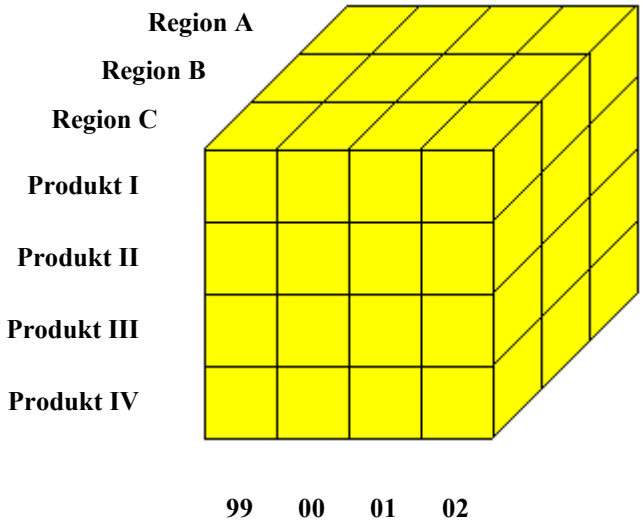


Abb. A.4.1: OLAP-Würfel mit den Dimensionen Region, Produkt und Zeit.
Quelle: Vgl. Schinzer, Bange, Mertens (1999), S. 40; eigene Darstellung.

4.2 Das Slice-Verfahren

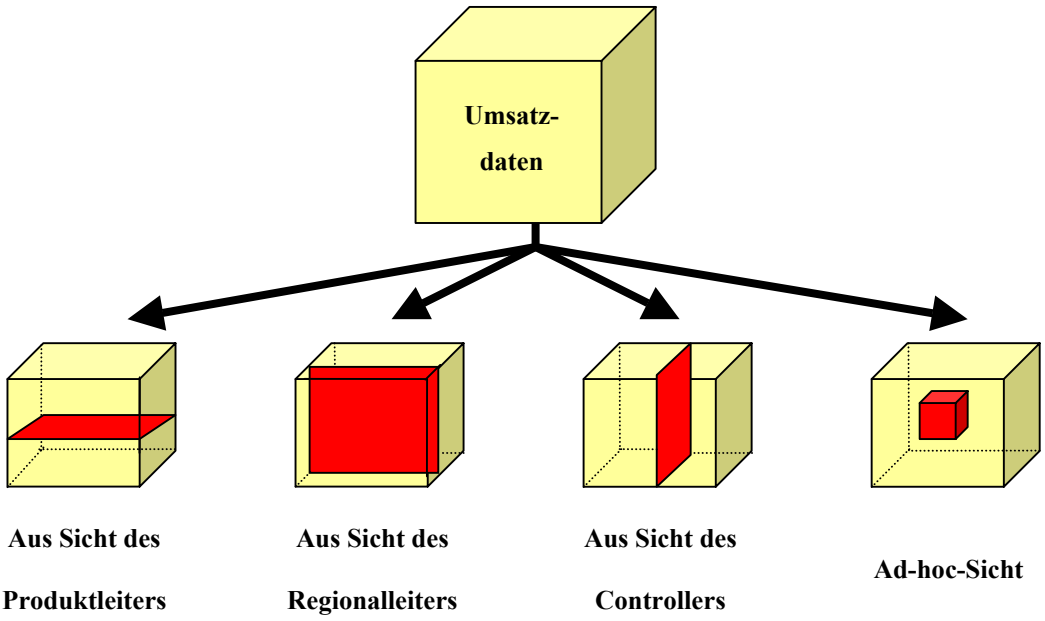


Abb. A.4.2: Selektion unterschiedlicher Datensichten mittels des Slice-Verfahrens.
Quelle: Vgl. Schinzer, Bange, Mertens (1999), S. 41; eigene Darstellung.

A.5 Einsatzgebiete des Data Mining

Funktionale Anwendungen

- **Marketing**
  - Segmentierung
  - Preisfindung
  - Database Management
  - Warenkorbanalyse

  ➔ *Customer Relationship Management*

- **Produktion**
  - Prozesssteuerung
  - Qualitätskontrolle
  - Bedarfsermittlung

  ➔ *Total Quality Management*
  ➔ *Six Sigma*

- **Controlling**
  - Deckungsbeitragsanalysen
  - Frühindikatoren

  ➔ *Balanced Scorecard*

- **Personal**
  - Personaleinsatzplanung
  - Personalbedarfsermittlung

- **Einkauf**
  - Qualitätssicherung
  - Identifizierung von Einsparungspotentialen
  - Procurement Vision

  ➔ *Supply Chain Management*
  ➔ *Supplier Relationship Management*

Branchenspezifische Anwendungen

- **Finanzdienstleistungen**
  - Kreditrisiko-Bewertung
  - Portfolio-Strategien
  - Kreditkartenmissbrauch
  - Schadensrisiko-Beurteilung

- **Handel**
  - Identifizierung von rentablen Stammkunden
  - Category Management
  - Warenkorbanalyse

- **Telekommunikation**
  - Kundenloyalitätsanalysen
  - Neukundengewinnung
  - Marktsegmentierung
  - Call Behavior Analyse

- **Gesundheitswesen**
  - Identifizierung erfolgreicher Therapien
  - Aufdeckung von Rentabilitätspotentiale bei verschreibungspflichtigen Präparaten
  - Aufdeckung von Betrugsversuchen

Abb. A.5: Einsatzgebiete des Data Mining: Funktionale und branchenspezifische Anwendungen.
Quelle: Eigene Darstellung.

A.6 Cross Validation

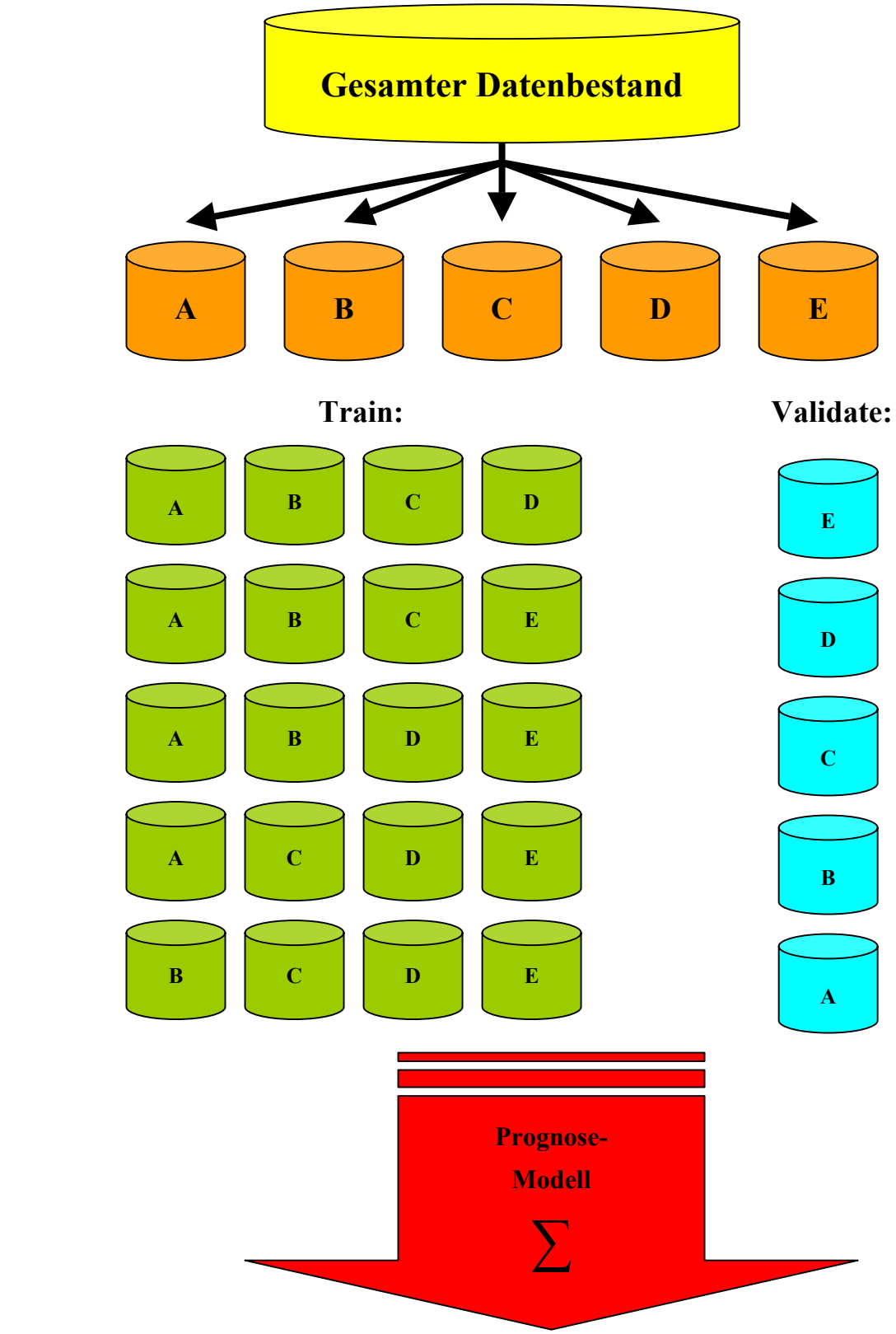


Abb. A.6: Cross Validation.
Quelle: Eigene Darstellung.

A.7 Complete Case Analysis

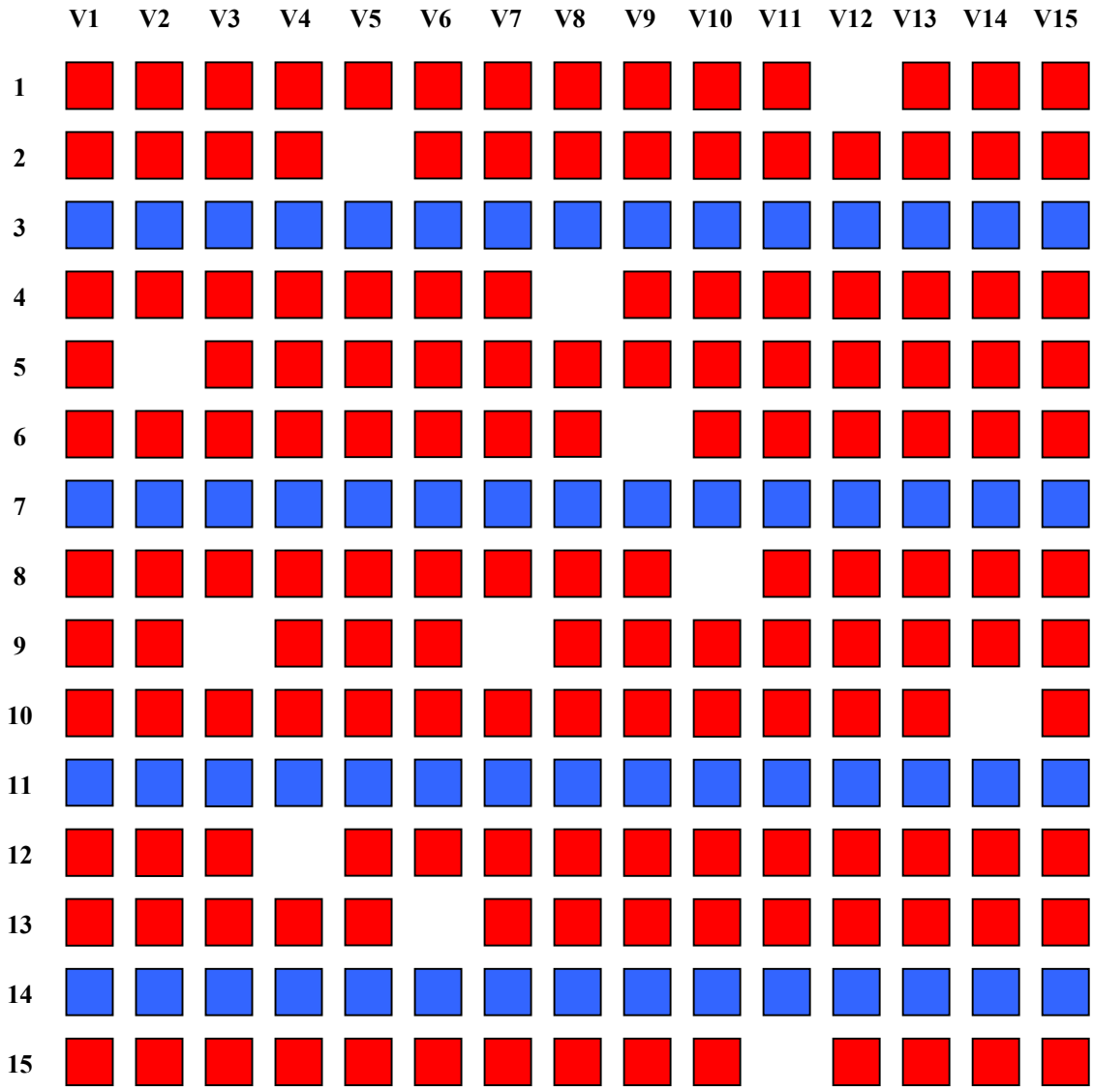


Abb. A.7: Complete Case Analysis.
Quelle: SAS Institute Inc. (2000d), S. 45; eigene Darstellung.

Abbildung A.7 verdeutlicht die Schwierigkeiten, die bei einer Complete Case Analysis auftreten können. Obwohl nur rund fünf Prozent der Daten fehlen, stehen 80 Prozent der Datensätze für die Modellanpassung nicht zur Verfügung. In diesem Fall würde auch keine Verbesserung erzielt, wenn einzelne Variablen entfernt würden, da jede Variable maximal einen fehlenden Wert enthält. Es gibt also keinen systematischen Zusammenhang zwischen dem Auftreten von Missing Values und den Variablen selbst.

A.8 Biologisches Neuron

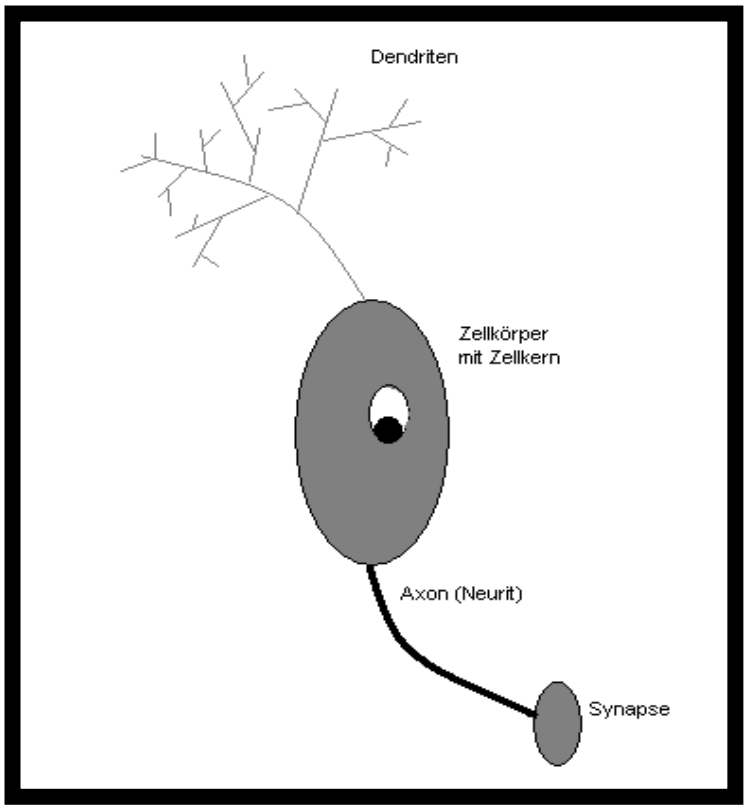


Abb. A.8: Biologisches Neuron.
Quelle: Vgl. Lämmel, Cleve (2001), S. 173; eigene Darstellung.

Dendriten nehmen die Erregungen auf, die von anderen Neuronen abstammen, und leiten sie an den Zellkern weiter. Die Erregung der Zelle wird dann über das Axon an die nächsten Neuronen weitergereicht. Die Synapsen stellen den Übergang vom Axon der einen Zelle zu den Dendriten weiterer Zellen dar. Ein biologisches Neuron hat mehrere tausend bis zehntausend Verbindungen zu weiteren Neuronen.

A.9 Aktivierungsfunktionen

9.1 Logistische Funktion

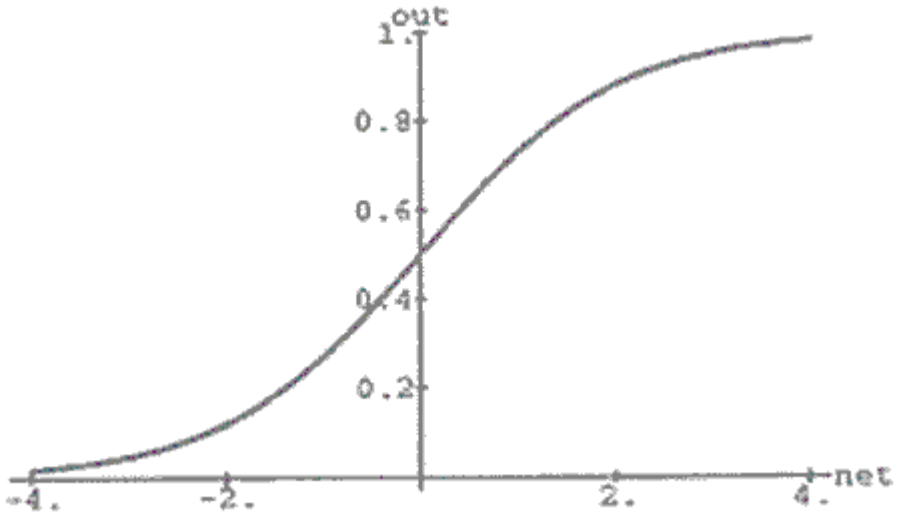


Abb. A.9.1: Logistische Funktion.
Quelle: Zell (2000), S. 77.

9.2 Tangens Hyperbolicus

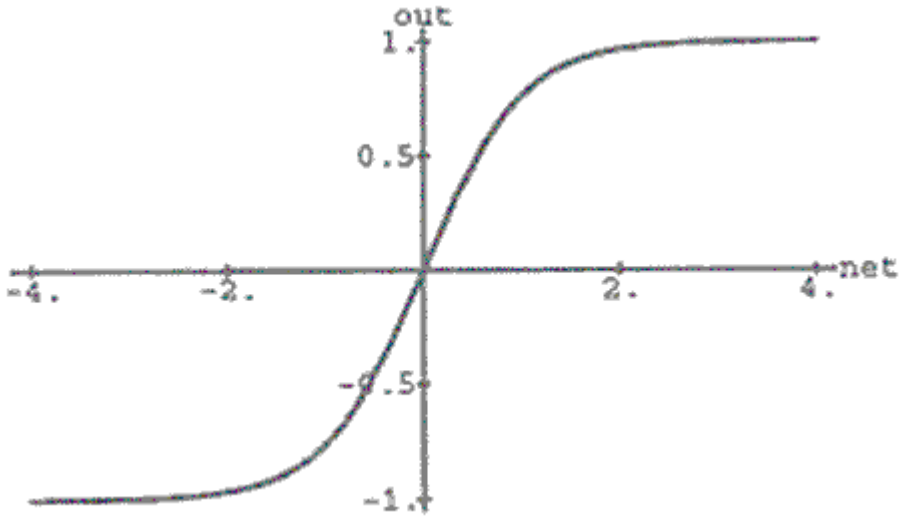


Abb. A.9.2: Tangens Hyperbolicus.
Quelle: Zell (2000), S. 77.

A.10 Topologien

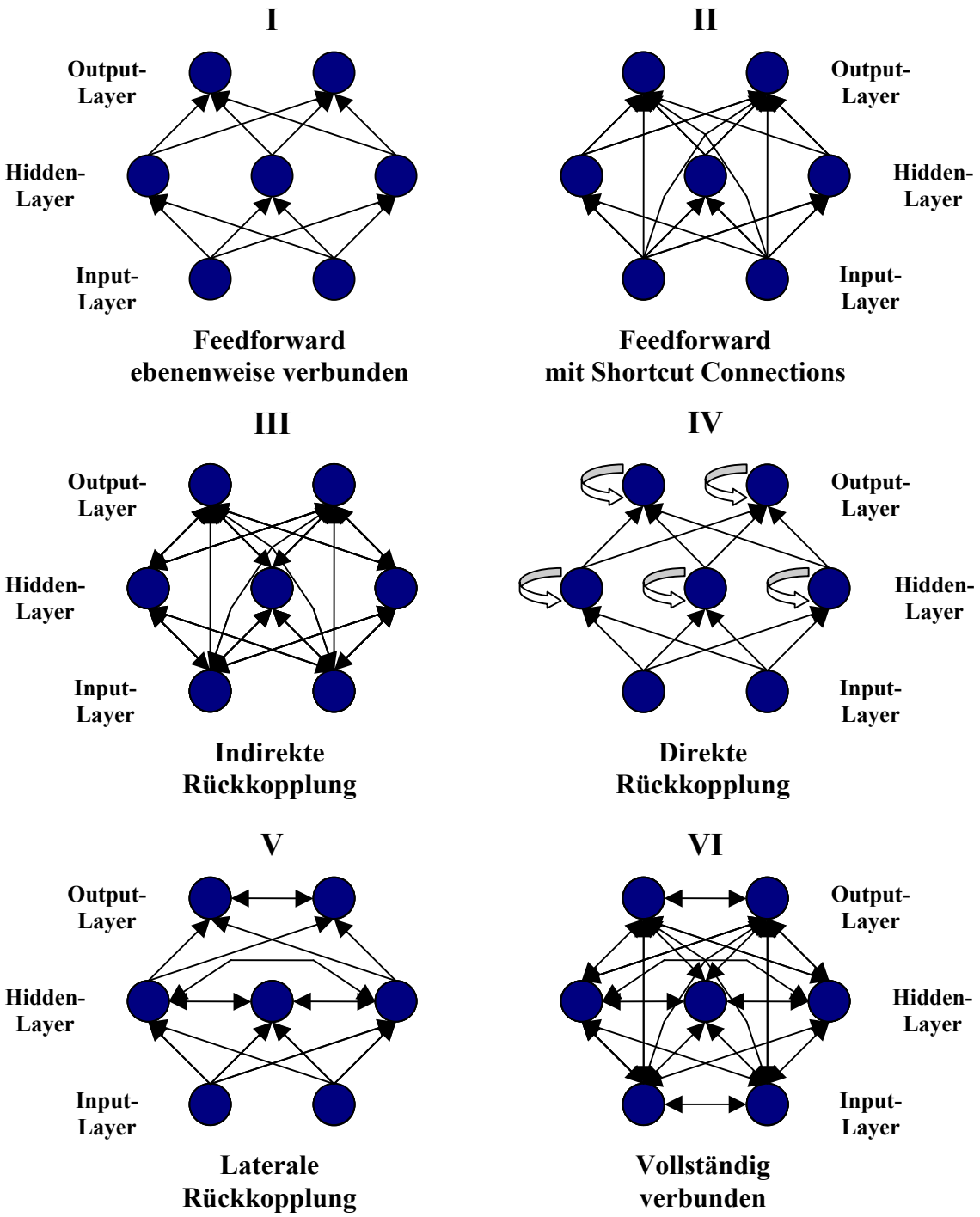


Abb. A.10: Topologien verschiedener KNN.
Quelle: Vgl. Zell (2000), S. 79; eigene Darstellung.

Die in Abbildung A.10 unter III bis VI gezeigten Netzwerkmodifikationen sind mögliche Anpassungen, die im *Neural Network*-Knoten des SAS® Enterprise Miner™ nicht vorgesehen sind. Dort sind als Topologien lediglich Feedforward-Netze (I) verankert, die ebenenweise verbunden sind. Außerdem gibt es die Möglichkeiten von Shortcut Connections (II).

A.11 Herleitung der Backpropagation-Regel

Wie in Abschnitt 4.4.3.2 beschrieben, ist der Ausgangspunkt für die Herleitung der Backpropagation-Regel das Gradientenverfahren, dass als einzelnes Argument dargestellt, folgende Form hat:

$$\Delta w_{ij} = -\eta \frac{\partial E(W)}{\partial w_{ij}} = \sum_p -\eta \frac{\partial E_p}{\partial w_{ij}} \quad \text{(A.7)}$$

Wie aus Gleichung (A.7) ersichtlich, existieren bezüglich der Gradientenverfahren zwei verschiedene Sichtweisen: das Batch-Verfahren, bei dem die Gewichtsänderungen erst nach der Bearbeitung des gesamten Satzes von Trainingsmustern werden oder das Online-Verfahren, das nach jedem einzelnen Muster die Verbindungsgewichte ändert. Die Betrachtung der Online-Variante ist von Vorteil, da auf die Kennzeichnung des Musters verzichtet werden kann.

Im Kontext der Gradientenverfahren ist ein Fehler die Abweichung der tatsächlichen Ausgabe von der erwarteten Ausgabe. Der Zusammenhang zwischen der Ausgabe und den Verbindungsgewichten ergibt sich aus der in Abschnitt beschriebenen Arbeitsweise eines Neurons:

(A.8) $o_j = f_{out}\,(f_{act}(net_j))$, mit $net_j = \sum_i o_i w_{ij}$ und f_{out} als *Identität*.

Die Ausgabe eines Neurons wird durch die Ausgabefunktion aus der Aktivierung des Neurons bestimmt. Die Aktivierung wird durch die Berechnung der Aktivierungsfunktion auf die Netzeingabe ermittelt. Die Netzeingabe ist abhängig von den Werten der Verbindungsgewichte. Nun kann Gleichung (A.7) durch Anwendung der Kettenregel auch folgendermaßen dargestellt werden:

$$\Delta w_{ij} = -\eta \underbrace{\frac{\partial E}{\partial o_j}}_{3.} \underbrace{\frac{\partial o_j}{\partial net_j}}_{2.} \underbrace{\frac{\partial net_j}{\partial w_{ij}}}_{1.} \quad \text{(A.9)}$$

Der Fehler ist nun abhängig vom Ausgabewert, der Ausgabewert ist abhängig von der Netzeingabe und diese wiederum von dem einzelnen Gewicht. Die Komponenten aus Gleichung (A.9) werden nun in umgekehrter Reihenfolge konkretisiert:

1. Anhand der Summationsformel für die Netzeingabe zeigt sich die Abhängigkeit der Netzeingabe bezüglich des Verbindungsgewichts w_{ij}. Durch die Ableitung entfallen alle Summanden, bis auf denjenigen mit k = i der w_{ij} enthält:

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_k o_k w_{kj} = o_i \quad \text{(A.10)}$$

2. Wird als Ausgabefunktion die Identität verwendet, ist die Abhängigkeit der Ausgabe von der Netzeingabe durch die Aktivierungsfunktion gegeben:

(A.11) $$\frac{\partial o_j}{\partial net_j} = \frac{\partial f_{act}(net_j)}{\partial net_j} = f'_{act}(net_j)$$

Dies gilt unter der Voraussetzung einer differenzierbaren Aktivierungsfunktion. Der Backpropagation-Algorithmus benutzt dafür die logistische Funktion[3] und erhält für die erste Ableitung:

(A.12) $$f_{Logistic}(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = f_{Logistic}(x)\cdot(1 - f_{Logistic}(x))$$

Und somit:

(A.13) $$\frac{\partial o_j}{\partial net_j} = f_{Logistic}(net_j)\cdot(1 - f_{Logistic}(net_j)) = o_j \cdot (1 - o_j)$$

3. Die Abhängigkeit des Fehlers vom Ausgabewert[4] wird als Fehlersignal δ definiert:

(A.14) $$\frac{\partial E}{\partial o_j} = \delta$$

Die Abhängigkeit des Fehlers von der Aktivität des Neurons kann nur für ein Ausgabeneuron direkt ausgedrückt werden. Als Fehlerfunktion wird die quadrierte Abweichung zuzüglich eines Faktors ½ verwendet[5].

(A.15) $$E = \frac{1}{2}\sum_j (t_j - o_j)^2 \text{ mit } \frac{\partial E}{\partial o_j} = \frac{\partial}{\partial o_j}\left(\frac{1}{2}\sum_k (t_k - o_k)^2\right) = -(t_j - o_j)$$

Zusammenfassend wird das Fehlersignal δ unter Verwendung der logistischen Funktion als Aktivierungsfunktion folgendermaßen bestimmt:

(A.16) $$\delta_j = \begin{cases} o_j(1-o_j)(t_j - o_j) & \text{falls j eine Ausgabezelle ist.} \\ o_j(1-o_j)\sum_k (\delta_k w_{jk}) & \text{falls j eine verdeckte Zelle ist.} \end{cases}$$

Damit wird das Fehlersignal eines Neurons einer inneren Schicht anhand der Fehlersignale aller nachfolgenden Zellen und der zugehörigen Verbindungsgewichte bestimmt. Folglich müssen zuerst die Fehlersignale der Ausgabeneuronen bestimmt, dann können die Fehlersignale der letzten inneren Schicht berechnet und schließlich alle vorherigen Fehlersignale bis zum ersten Neuron der inneren Schicht iterativ bestimmt werden.

---

[3] Alternative Aktivierungsfunktionen sind der Tangens Hyperbolicus oder die Identität. Schwellenwertfunktionen können aufgrund ihrer nicht vorhandenen Differenzierbarkeit nicht verwendet werden.
[4] Als Ausgabefunktion wird die Identität benutzt.
[5] Auf diese Weise wird verhindert, dass sich positive und negative Abweichungen gegeneinander aufheben. Der Faktor dient lediglich zur Vereinfachung der Ableitung.

A.12 Support, Konfidenz und Lift einer Assoziationsregel

Szenario:

		Produkt B: Nein	Produkt B: Ja
Produkt A	Nein	500	3500
	Ja	1000	5000

Support:

Anteil der Käufer die Produkt A und B zusammen kaufen.

Konfidenz:

Anteil der Käufer von Produkt A, die auch Produkt B kaufen.

Lift:

Korrelation zwischen Produkt A und B.

Regel: (Produkt A ➔ Produkt B)

Support: $$\frac{A+B}{Gesamt} = \frac{5000}{10.000} = 0{,}5 = 50\ \%$$

Konfidenz: $$\frac{A+B}{A} = \frac{5000}{6000} = 0{,}83 = 83\ \%$$

Lift: $$\frac{\frac{A+B}{A}}{\frac{B}{Gesamt}} = \frac{\frac{5000}{6000}}{\frac{8500}{10.000}} = \frac{0{,}83}{0{,}85} < 1$$

Abb. A.11: Support, Konfidenz und Lift einer Assoziationsregel (Produkt A ➔ Produkt B).
Quelle: SAS Institute Inc. (2002), S. 8-5; eigene Darstellung.

A.13 ROC-Kurve

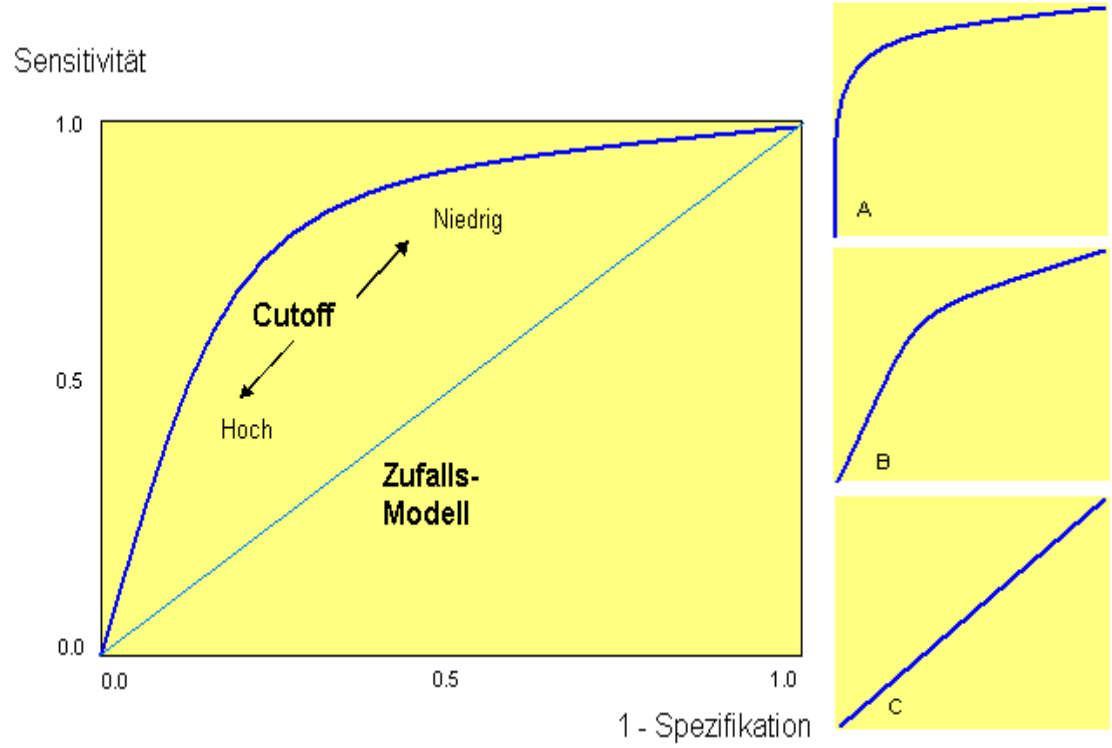


Abb. A.12: ROC-Kurve.
Quelle: Screenshot SAS® System Help; eigene Darstellung.

Alle Punkte auf der ROC-Kurve zeigen einen möglichen Cut-Off. Die Wahl des Cut-Offs beschreibt einen Trade-Off zwischen Sensitivität und Spezifikation. Die Fläche unter der Kurve[6] ist ein Maß für die diskriminatorische Aussagekraft des Modells bei verschiedenen Cut-Offs. Ein Modell ohne Trennkraft ist das Zufallsmodell, dargestellt durch die 45°-Linie. Die Kurven A, B, und C zeigen demnach unterschiedlich gute Modelle. So zeigt Kurve A ein sehr gutes Vorhersagemodell. Kurve C zeigt hingegen ein schlechtes Klassifikationsmodell, die gesamte Bandbreite der Cut-Offs besitzt keine Trennkraft. Dies bewirkt eine schlechte Klasseneinteilung.

Ein optimaler Cut-Off lässt sich folgendermaßen ermitteln:

$$(A.17)\ \zeta_{opt} = \frac{1}{1+\frac{\delta_{TP}-\delta_{FN}}{\delta_{TN}-\delta_{FP}}}$$

[6] Je dichter die Kurve an der Nordwest-Ecke liegt, desto besser.

A.14 Die SEMMA-Methode

Die einzelnen Icons oder „Werkzeuge“ des SAS® Enterprise Miner™, auch Knoten genannt, sind in der Reihenfolge des Data Mining-Prozesses angeordnet. Die Verfahren und Analyseschritte werden in fünf Gruppen unterteilt, aus denen das Schlagwort SEMMA entsteht.

SEMMA setzt sich zusammen aus:

Sample: Die optionale Stichprobenbildung und das Bereitstellen von Trainings-, Validierungs- und Testdaten.

Explore: Die Exploration der Daten inklusive Variablenauswahl, Gruppierung und Visualisierung. Dies beinhaltet auch die Möglichkeit einer Assoziationsanalyse.

Modify: Die Daten werden für die Analyse mit den Data Mining-Methoden modifiziert und transformiert, außerdem stehen Verfahren zur Klassifizierung bereit.

Model: DieVorhersage- bzw. Klassifikationsmodelle (z.B.: Regressionsanalyse, Entscheidungsbaumverfahren oder künstliche neuronale Netze) werden entwickelt und durchgeführt.

Assess: Die verschiedenen Verfahren werden miteinander verglichen, um die beste Performance für das Problem zu gewährleisten.

Des Weiteren sind zusätzliche Tools zur Vereinfachung der Prozessdiagramme vorhanden und die Möglichkeit implementiert eine Data Mining Datenbank zu erstellen.

A.15 Eine kurze Erläuterung des SAS® Enterprise Miner™

Die Oberfläche des SAS® Enterprise Miner™ besteht im Wesentlichen aus zwei Fenstern.

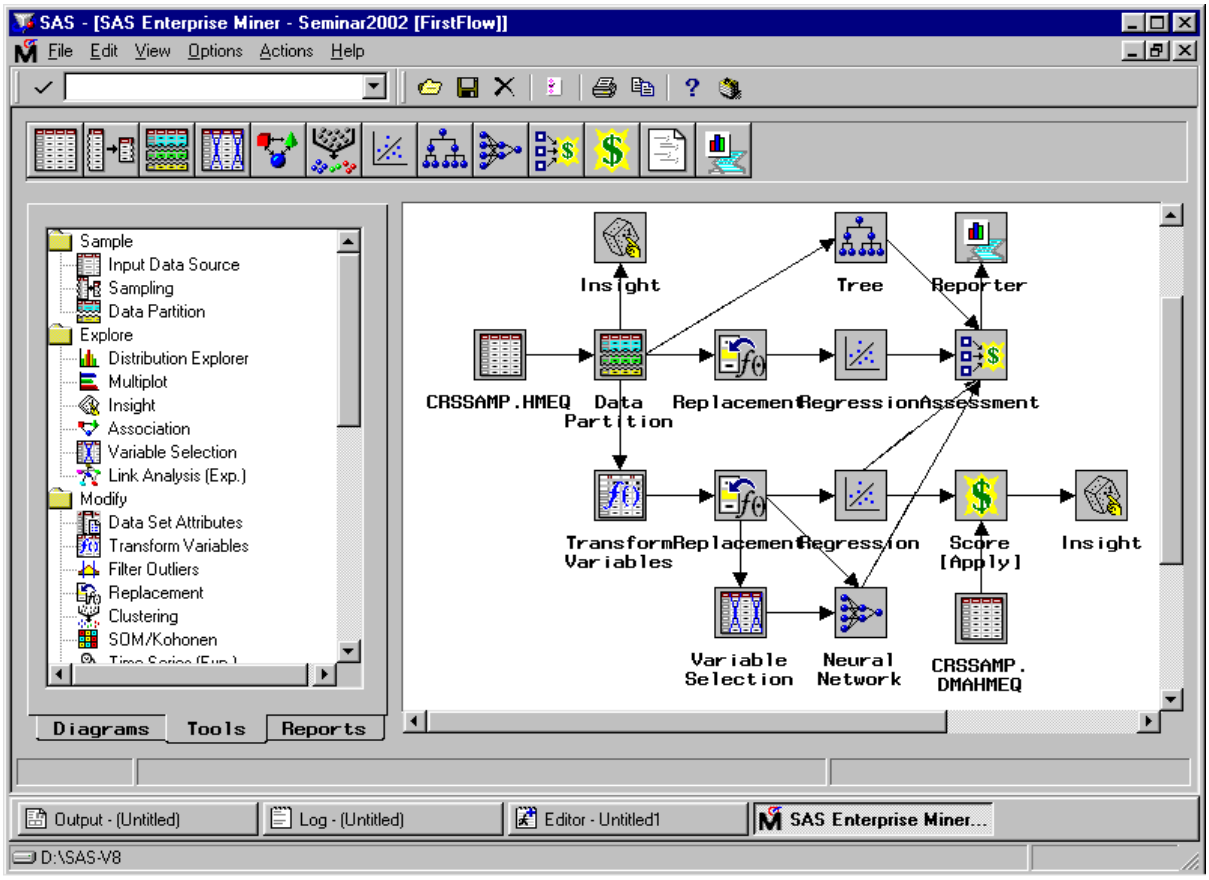


Abb A.13: Die Benutzeroberfläche des SAS® Enterprise Miner™.
Quelle: Screenshot SAS® Enterprise Miner™.

Im linken Fenster wird unter *Diagrams* die Projekt- und Diagrammverwaltung durchgeführt, unter *Tools* stehen die gesamten zum Data Mining mit der SEMMA-Methode benötigten Werkzeuge zur Verfügung und unter *Reports* können Berichte verwaltet werden. Das rechte Fenster ist für den eigentlichen Data Mining-Prozess, die Fläche für die Umsetzung. Hier werden die Icons per Drag and Drop abgelegt und durch Pfeile verbunden, auf diese Weise entstehen Knoten. Wird auf den Knoten die rechte Maustaste gedrückt, öffnet sich ein Pop-Up-Fenster. Hier stehen verschiedene Möglichkeiten zur Verfügung. Der Befehl *Open* ermöglicht den Zugang zu dem jeweiligen Knoten. Nun kann man aus der umfassenden Anzahl an Optionen wählen und den Knoten an das jeweilige Problem anpassen. Ist dieser Prozess abgeschlossen, führt der nächste Befehl *Run* im Pop-Up-Fenster, analog zu dem ansonsten in SAS® System verwendeten *Submit*, zur Ausführung des Knoten. Eine grüne Umrandung verrät den Fortschritt des gerade bearbeiteten Prozesses. Die Ergebnisse können unter *Results* angesehen werden. Findet eine Modellierung statt, wird eine weitere Option, *Model Manager,* hinzugefügt.

A.16 Die Knoten der Sample-Gruppe

Abb. A.14: Die Knoten der Sample-Gruppe: Input Data Source, Sampling und Data Partition.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

16.1 Input Data Source

Die Daten für die KDD-Analyse werden mit dem *Input Data Source*-Knoten eingelesen, bereitgestellt werden sie i.d.R. mit Hilfe der SAS® Warehouse Administrator-Software™, aber auch große Tabellen oder Views, die in denormalisierter Form vorliegen, können für die Analyse verwendet werden. Die Daten sollten, um eine gute Generalisierungsfähigkeit zu gewährleisten, regelmäßig aktualisiert werden. Die Daten sollten hierzu in einer niedrigen Granularität vorliegen. Ein hoher Detaillierungsgrad, vorbereinigte und weitgehend vollständige Daten mit einer hohen Anzahl an Attributen sind die ideale Grundlage für Data Mining.

Automatisch wird anhand der Quelldaten eine statistische Auswertung mitgeliefert und die Metadaten für die einzelnen Variablen definiert. Metadaten sind beschreibende Informationen über die Struktur und den Inhalt der Daten und über die Applikationen und Prozesse, die auf diese Daten zugreifen. Die Prädikatsmerkmale (Measurement Level) und der Einfluss, den die Variable in der Analyse einnehmen soll (Model Role), können geändert werden. Für die Ziel- oder Target-Variablen lassen sich sog. Target-Profile definieren, so dass der potentielle Return On Investment (ROI) und der entstehende Gewinn in der Auswertung abzulesen sind.

16.2 Sampling und Data Partition

Der *Sampling*-Knoten findet bei einer extrem umfangreichen Datenbasis Verwendung. Durch Stichproben wird eine signifikante Verkürzung der Arbeitsprozesse erreicht (vgl. Abschnitt 3.1.3.2). Die Einteilung der Daten in den Trainings-, Validierungs- und Test-Prozess wird im *Data Partition*-Knoten vorgenommen (vgl. Abschnitt 3.1.1).

A.17 Die Knoten der Explore-Gruppe

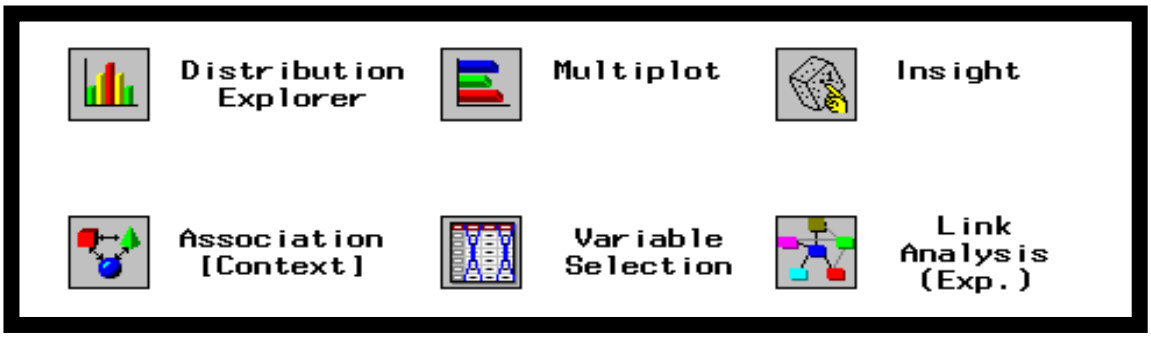


Abb. A.15: Die Knoten der Explore-Gruppe: Distribution Explorer, Multiplot, Insight, Association, Variable Selection und Link Analysis.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

17.1 Distribution Explorer, Multiplot und Insight

Diese Knoten ermöglichen die Visualisierung großer Datenvolumina. Im *Distribution Explorer*-Knoten sind Verteilungen in multidimensionalen Histogrammen mit bis zu drei Variablen möglich. Bar Charts werden im *Multiplot*-Knoten verwendet. Außerdem lassen sich für die Variablen die Anteile an der Target-Variable anzeigen. Der *Insight*-Knoten bietet Zugang zur SAS® INSIGHT™-Software.

17.2 Association

Mit dem *Association*-Knoten lassen sich Beziehungen mit der Assoziations- und Sequenz-Analyse aufdecken. Diese Verfahren werden in Abschnitt 2.2.1.3 eingeführt und ab Abschnitt 4.5 ausführlich besprochen.

17.3 Variable Selection

Der *Variable Selection*-Knoten bestimmt die Bedeutung der Regressoren auf die zu prognostizierende oder klassifizierende Zielvariable. Für die Selektion werden entweder das R^2- oder das χ^2 Kriterium verwendet. Variablen werden abgewiesen, wenn keine Korrelation zum Zielwert besteht oder der Anteil an Missing Values zu groß ist (vgl. Abschnitt 3.2).

17.4 Link Analysis

Die Link Analyse nutzt die Graphentheorie um Verbindungen und Beziehungen zwischen Variablen und deren Auswirkungen zu veranschaulichen. Komplexe Daten werden so mit Hilfe eines Graphen dargestellt. Die Link Analyse kann sowohl als Verfahren zur Dimensionsreduktion als auch zur Segmentierung von Variablen verstanden werden.

A.18 Die Knoten der Modify-Gruppe

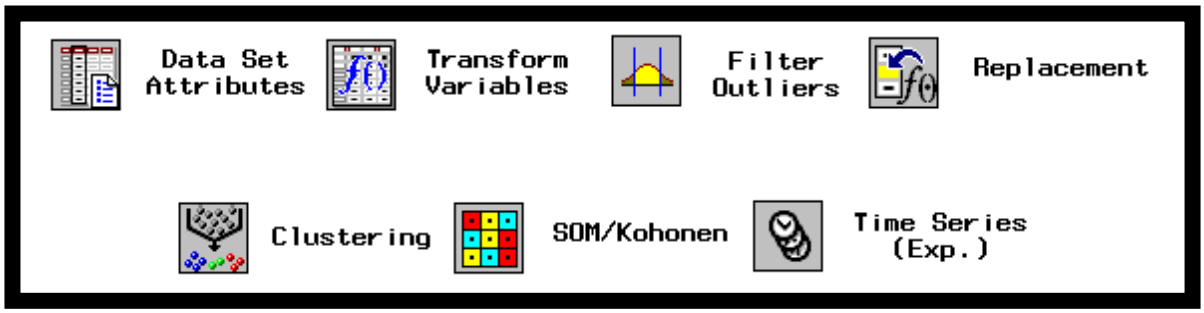


Abb. A.16: Die Knoten der Modify-Gruppe: Data Set Attributes, Transform Variables, Filter Outliers, Replacement, Clustering, SOM / Kohonen und Time Series.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

18.1 Data Set Attributes

Veränderungen der Attribute des Datensatzes lassen im *Data Set Attributes*-Knoten vornehmen. Die Metadaten, das Measurement Level, die Model Role, Beschreibungen und Namen der Attribute sind modifizierbar, außerdem lassen sich hier Target-Profile erstellen.

18.2 Transform Variables und Filter Outlier

Verfahren wie die Regressionsanalyse oder KNN verlangen normalverteilte Variablen. Der *Transform Variables*-Knoten nutzt verschiedene Transformationen wie Natürlichen Logarithmus, Potenzen, Normalisierungen oder die Korrelation zur Zielvariable maximieren. Zusätzlich können benutzerdefinierte Formeln erstellt werden. Eine gängige Methode ist das Gruppieren über Quantile oder Schranken. Ausreißer und Extremwerte werden im *Filter Outlier*-Knoten behandelt. Allerdings findet diese Anwendung nur bei den Trainingsdaten statt, um an dieser Stelle eine bestmögliche Anpassung zu erzeugen (vgl. Abschnitt 3.3).

18.3 Clustering und SOM / Kohonen

Mit diesen Knoten lassen sich Segmentierungen durchführen. Die Verfahren werden in den Abschnitten 2.2.1.1.2 und 2.2.1.2.3 beschrieben, eine detaillierte Analyse findet sich dann ab den Abschnitten 4.2 und 4.4.5.

18.4 Time Series

Der *Time Series*-Knoten ermöglicht eine Zeitreihenanalyse um Trends und saisonale Komponenten zu identifizieren, die der Generalisierungsfähigkeit entgegenwirken. In unregelmäßigen Zeitabständen gesammelte, zeitpunktbezogene Daten werden über die Zeit mit einer gewissen Frequenz zusammengefasst.

A.19 Die Knoten der Model-Gruppe

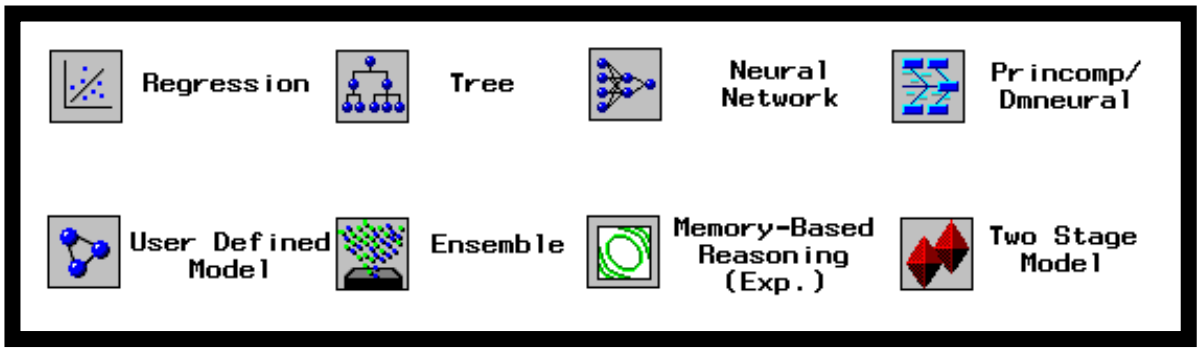


Abb. A.17: Die Knoten der Model-Gruppe: Regression, Tree, Neural Network, Princomp / Dmneural, User Defined Model, Ensemble, Memory-Based Reasoning und Two Stage Model.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

19.1 Regression, Tree und Neural Network

Mit diesen Knoten werden Regressionsanalysen und Entscheidungsbaumverfahren durchgeführt sowie künstliche neuronale Netze konstruiert. Diese Verfahren sind der Hauptbestandteil dieser Arbeit. Eine charakteristische Beschreibung wird in den Abschnitten 2.2.1.1.1, 2.2.1.2.1 und 2.2.1.2.2 vorgenommen, eine detaillierte Analyse findet sich dann ab den Abschnitten 4.1, 4.3 und 4.4.

19.2 Princomp / Dmneural

Der *Princomp / Dmneural*-Knoten ermöglicht die Durchführung einer Hauptkomponenten-Analyse, die ohne Target-Variable auskommt. Außerdem können Modellanpassungen an zusammengesetzte nicht-lineare Modelle durchgeführt werden. Dafür werden die Hauptkomponenten als Inputs verwendet, die ein binäres oder intervallskaliertes Ziel vorhersagen.

19.3 User-Defined Model

Der *User-Defined Model*-Knoten ermöglicht die Entwicklung eigener Modelle durch das Einfügen von SAS®-Codes. Hierfür kann die umfangreiche SAS® STAT™-Software genutzt werden. Außerdem können Werte, die dem Zielwert entsprechen, in einen Datensatz gespeichert werden, um wiederum nach Eingabe in den *Input Data Source*-Knoten untersucht werden. So lassen sich z.B. von den „guten“ Werten die Herausragenden separieren.

19.4 Ensemble

Der *Ensemble*-Knoten bietet folgende Möglichkeiten: Kombinierte Modelle, stratifizierte Modelle und Bagging- bzw. Boosting-Modelle.

1. Verschiedene Modelle werden generiert, ausgehend von dem gleichen *Input Data Source*-Knoten, identischer Partitionierung und eventuellen Transformationen. Der *Ensemble*-Knoten bildet nun den Durchschnitt der Ereigniseintrittswahrscheinlichkeiten (Class-Target) oder der prognostizierten Werte (Intervall-Target). Das Ensemble-Modell wird dann zu einem einzelnen Modell.
2. Stratifizierte bzw. gruppierte Variablen können in Modelle integriert werden, die für jedes Segment mit IF-THEN-DO / END-Blöcken kombiniert werden und so in den Score-Code einfließen. Dafür wird der *Group Processing*-Knoten (vgl. Kapitel 22 Abschnitt 22.3) mit der Einstellung *Variables* vor das jeweilige Modell geschaltet. Diese Technik ist im Zusammenhang mit gemischten Populationen von Vorteil.
3. Bagging- bzw. Boosting-Modelle werden in Abschnitt 4.3.4 im Zusammenhang mit dem Entscheidungsbaumverfahren erläutert, sind aber grundsätzlich auch mit allen anderen Verfahren kombinierbar.

19.5 Memory-Based Reasoning

Memory-Based Reasoning oder fallbasiertes Schließen identifiziert ähnliche Fälle und nutzt diese Informationen für neue Daten. Für die Kategorisierung oder Vorhersage von Beobachtungen wird der K-Nearest Neighbor-Algorithmus angewandt.

19.6 Two Stage Model

Das Two Stage Model wird angewendet, wenn eine Class und eine Intervall Variable als Target-Variablen fungieren. Dies ist vor allem dann interessant, wenn außer der Klassifikation einer Eintrittswahrscheinlichkeit, das korrespondierende Ausmaß vorhergesagt werden soll.

A.20 Die Knoten der Assess-Gruppe

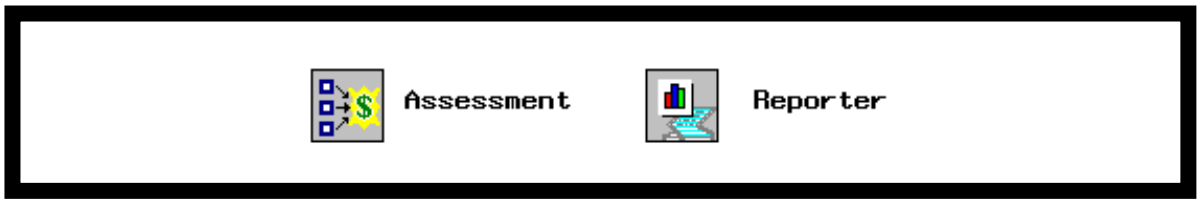


Abb. A.18: Die Knoten der Assess-Gruppe: Assessment und Reporter.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

20.1 Assessment

Der *Assessment*-Knoten bietet den Rahmen um konkurrierende Modelle gegenüberzustellen. Der Vergleich basiert auf dem erwarteten und aktuellen Gewinn bzw. Verlust. Mit verschiedenen Grafiken wie Lift Charts oder Profit / Loss Charts werden die Nützlichkeit der Modelle beschrieben. Außerdem werden die evt. definierten Target-Profile visualisiert. Eine weitere Option ist die ROI-Analyse. Darüber hinaus wird eine statistische Zusammenfassung mit den Kriterien BSC, Root ASE und Misclassification Rate angeboten und eine Confusion-Matrix gezeigt (vgl. Kapitel 5)

20.2 Reporter

Eine Zusammenfassung der Ergebnisse wird im *Reporter*-Knoten angeboten, die mit einem Web-Browser angezeigt werden können. Jeder Report beinhaltet ein Prozess-Flussdiagramm und Berichte zu jedem Knoten. Die Reports-Spalte in der Oberfläche des SAS® Enterprise Miner™ ist die dafür vorgesehene Verwaltungsebene.

A.21 Die Knoten der Score-Gruppe

Abb. A.19: Die Knoten der Score-Gruppe: Score und C*Score.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

21.1 Score

Die Generalisierung trainierter Modelle auf neue Daten wird im *Score*-Knoten vorgenommen. Die Ergebnisse werden auch in Form eines SAS®-Codes geliefert, damit auch ohne SAS® Enterprise Miner™ in der SAS®-Umgebung neue Daten mittels des trainierten Modells generalisiert werden können.

*21.2 C*Score*

Der SAS®-Code des *Score*-Knoten kann im *C*Score*-Knoten in die C Programmiersprache übersetzt werden.

A.22 Die Knoten der Utility-Gruppe

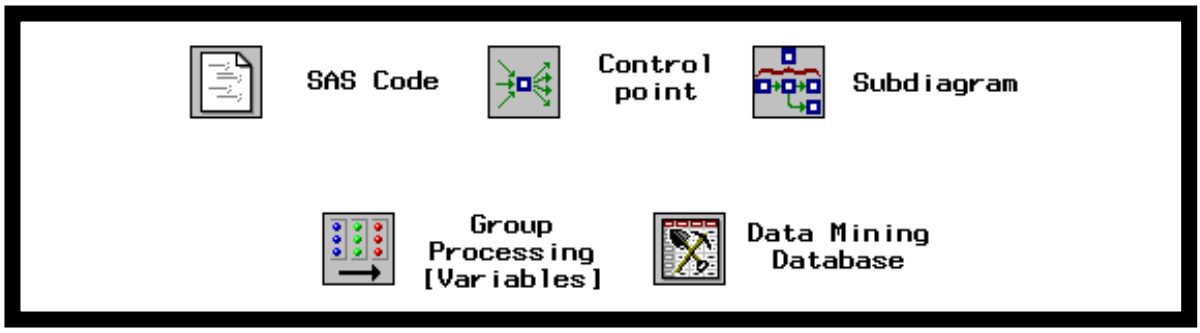


Abb. A.20: Die Knoten der Utility-Gruppe: SAS Code, Control Point, Subdiagram, Group Processing und Data Mining Database.
Quelle: Screenshot SAS® Enterprise Miner™; eigene Darstellung.

22.1 SAS Code

Der *SAS Code*-Knoten erweitert die Funktionalität des SAS® Enterprise Miner™. Die im SAS®-System vorhandenen Prozeduren werden in die Data Mining-Analyse integriert.

22.2 Control Point und Subdiagram

Der Control Point sorgt für Übersichtlichkeit und vereinfacht oftmals Strukturen. Häufig lassen sich, sollten mehrere Knoten vernetzt werden, so ein Reduzierung der Verbindungen erreichen. Eine weitere Vereinfachung lässt sich durch den *Subdiagram*-Knoten erzielen. Einzelne Diagrammteile lassen sich so durch einen einzigen Knoten ersetzen, wobei die Struktur nach Öffnung immer noch zu betrachten ist.

22.3 Group Processing

Der *Group Processing*-Knoten wird benötigt, um die im *Ensemble*-Knoten (vgl. Kapitel A.19 Anschnitt 19.4) besprochenen, stratifizierten, Bagging- und Boosting-Modelle zu verwirklichen. Ein weiteres Einsatzgebiet ist das Two Stage Model (vgl. Kapitel A.19 Abschnitt 19.6). Außerdem können Modelle wiederholt durchgeführt werden.

22.4 Data Mining Database

Eine Datenbank für den Data Mining-Prozess lässt sich im *Data Mining Database*-Knoten erstellen. Hierdurch lässt sich die Datenbank auf die Anforderungen abstimmen und so eine Performanzoptimierung erreichen. Ein weiterer Vorteil ist die eventuelle Reduzierung von Analyseschritten. Die Datenbank enthält einen Metadaten-Katalog, eine statistische Auswertung für intervallskalierte Variablen und Informationen über die Class-Variablen.

A.23 Prozessbeschreibungen der Fallstudien - Fallstudie A

Eine Variablen-Übersicht, die angibt, welche Skalierung, Modellrolle und Beschreibung die Variablen in dem verwendeten Beispiel besitzen, wird nach der Prozessbeschreibung präsentiert.

1. Neues Projekt anlegen:

➔ **File ➔ New ➔ Project**

Projektname: *Mailing*; Diagrammname: *Dining.*

Folgendes SAS® Enterprise Miner™-Diagramm ist nötig zur Bearbeitung der Fallstudie:

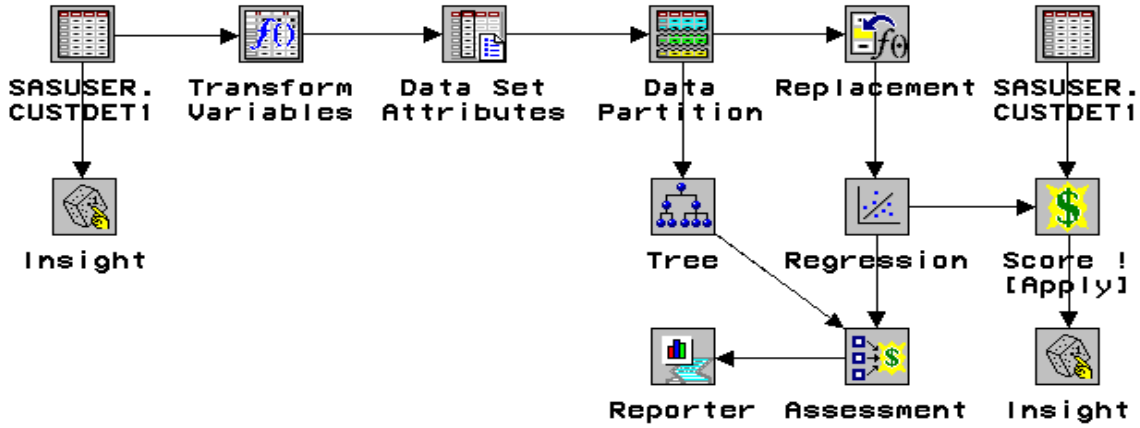


Abb. A.21: SAS® Enterprise Miner™-Diagramm für Fallstudie A.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source: Datensatz einlesen:

➔ Rechter Mausklick ➔ **Open** ➔ **Select** ➔ **SASUSER**-Library und **CUSTDET1**-Datensatz auswählen ➔ Speichern und schließen.

3. Insight: Datensatz auswerten:

➔ Rechter Mausklick ➔ **Open** ➔ **Entire data set**-Button auswählen ➔ Speichern und schließen.

➔ Rechter Mausklick ➔ **Run** ➔ **View Results: Yes**

An dieser Stelle erscheint der komplette Datensatz des Beispiels. Außerdem sind folgende Analysen möglich:

➔ **Analyze:**

Histogram / Bar Chart (Y), Box Plot (Y), Line Plot (YX), Scatter Plot (YX), Contour Plot (ZYX), Rotating Plot (ZYX), Distribution (Y), Fit (YX) und Multivariate (YX).

4. Transform Variables: Target-Variable erzeugen

Aus den Variablen **KITCHEN**, **DISHES** und **FLATWARE** soll die neue Zielvariable **DINEBIN** erzeugt werden. Außerdem muss eine Transformation von intervallskalierten zu einer binären Variable erfolgen.

➔ Rechter Mausklick ➔ **Open** ➔ **Create Variable**-Icon anklicken ➔ Name: *DINEBIN*; Label: *DINING No / Yes* ➔ **Define** ➔ In das Formelfeld **DINEBIN(N)**: *dining > 0* ➔ **OK** ➔ **OK** ➔ Speichern und schließen.

5. Data Set Attributes: Bestimmung der Zielvariable, des Target-Profils und Prior-Vektors

Zielvariable identifizieren:

➔ Rechter Mausklick ➔ **Open** ➔ Variable **DINEBIN** auswählen ➔ Rechter Mausklick auf Spalte **New Model Role** ➔ **target** auswählen.

Die Variablen **KITCHEN**, **DISHES** und **FLATWARE** werden in der Variable **DINING** zusammengefasst, die dann schließlich zur binären Target-Variable **DINEBIN** wird:

➔ Variable **DINEBIN** auswählen ➔ Rechter Mausklick auf Spalte **New Measurement** ➔ **binary** auswählen.

Abschließend werden die vorherigen Variablen von der Analyse ausgeschlossen:

➔ Variablen **DINING, KITCHEN, DISHES** und **FLATWARE** auswählen ➔ Rechter Mausklick auf Spalte **New Model Role** ➔ **rejected** auswählen.

Die Zielvariable sollte absteigend sortiert sein, da die Werte mit der Ausprägung 1 identifiziert werden sollen (1 = Kauf):

➔ Im **Class Variables**-Tab die Variable **DINEBIN** auswählen ➔ Rechter Mausklick auf Spalte **New Order** ➔ **Set New Order** ➔ **Descending** auswählen.

Target-Profil definieren:

➔ Im **Variables**-Tab die Variable **DINEBIN** auswählen ➔ Rechter Mausklick auf Spalte **New Model Role** ➔ **Edit target profile** ➔ **Yes** ➔ Im **Assessment Information-**Tab rechter Mausklick in das linke Dialog-Fenster ➔ **Add** ➔ **Profit matrix** auswählen ➔ **Edit Decisions** anklicken und **Maximize profit with costs** auswählen.

In der Reihe mit der Entscheidung Kauf = **1** werden die konstanten Kosten eingetragen: **COST** = *10;* in der Matrix wird in Reihe und Spalte **1** der Wert *90* eingetragen und in der Reihe und Spalte **0** der Wert *0*.

Rechter Mausklick auf **Profit matrix** ➔ **Set to use** auswählen ➔ speichern.

Prior-Vektor definieren:

➔ Im **Variables**-Tab rechter Mausklick in das linke Dialog-Fenster ➔ **Add** ➔ Für **Level 1** = *0.12* und für **Level 0** = *0.88* eintragen ➔ Rechter Mausklick auf **Prior vector**

➔ **Set to use** auswählen ➔ Speichern ➔ Speichern und schließen.

6. Data Partition: Einteilung in Trainings-, Validierungs- und Testdaten

➔ Rechter Mausklick ➔ **Open** ➔ Im Feld **Train** = *50%*, im Feld **Validation** = *25%* und im Feld **Test** = *25%* eintragen ➔ Speichern und schließen.

7. Replacement: Fehlende Werte ersetzen

Als Einfügestrategie werden Indikator-Variablen verwendet:

➔ Rechter Mausklick ➔ **Open** ➔ **Create imputed indicator variables** anklicken

➔ Speichern und schließen

8. Regression: Anpassung an eine Regressionsmodell

➔ Rechter Mausklick ➔ **Open** ➔ Im **Selection Method**-Tab **Stepwise** auswählen

➔ Speichern und schließen.

Model Name: *StepReg*; **Model Description:** *Stepwise Logistic Regression.*

9. Decision Tree: Anpassung an ein Entscheidungsbaummodell

➔ Rechter Mausklick ➔ **Open** ➔ Im **Basic**-Tab als **Splitting criterion** den **Chi-square test** auswählen ➔ Speichern und schließen.

Model Name: *ChiTree*; **Model Description:** *Chi-squared decision tree.*

10. Assessment: Modellbewertung

Zuerst wird die Data Mining-Analyse gestartet. Alle Knoten (von der Input Data Source bis zum Assessment) werden bearbeitet:

➔ **Run** ➔ **View Results: Yes**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang:

➔ **Tree-** und **Regression-**Model auswählen ➔ **Tools** ➔ **Lift Chart**

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

11. Reporter: Erstellung eines HTML-Berichts

➔ Rechter Mausklick ➔ **Run**

Mit einem Web Browser lassen sich nun die Berichte öffnen:

12. Score: Score-Code exportieren

Da das Regressionsmodell in der Assessment-Bewertung ausgesucht wurde, wird dieses nun mit dem Score-Knoten verbunden:

➔ Rechter Mausklick ➔ **Open** ➔ Im **Settings**-Tab den Schalter **Inactive** auswählen.

Als nächster Schritt muss der Score-Code gespeichert werden:

➔ Im **Score code-**Tab im linken Dialog-Fenster das Regressionsmodell auswählen

➔ Rechter Mausklick ➔ **Save** ➔ **Name:** *Logistic Regression Code* ➔ **OK.**

Nun wird der Code exportiert, damit er auch im SAS®-System im **Program Editor** ausgeführt werden kann:

➔ Rechter Mausklick auf **Logistic Regression Code** ➔ **Export** ➔ **Save As**

➔ **Name:** *REGRESS.SAS* ➔ Speichern und schließen

Folgender Code muss zusätzlich eingeführt werden, um die Zielvariable zu bestimmen und um die Datei, samt Library, in der die Daten zu finden sind, anzugeben:

```
%let _PREDICT=Variable;
%let _SCORE=Library.Dateiname;
```

13. Input Data Source: Auswahl des zu scorenden Datensatzes

➔ Rechter Mausklick ➔ **Open** ➔ **SASUSER.CUSTDET1** auswählen ➔ Die **Role** der Daten von **RAW** auf **SCORE** setzen.

➔ Speichern und schließen.

14. Score: Daten scoren

➔ Rechter Mausklick ➔ **Open** ➔ **Apply training data score code to score data set** auswählen.

➔Speichern und schließen.

15. Insight: Gescorte Daten auswerten

➔ Rechter Mausklick ➔ **Open** ➔ **Select**-Button neben dem **Data set**-Feld auswählen.

Nun muss der Datensatz ausgewählt werden, der für den Score-Vorgang zur Verfügung gestellt wird. Typischerweise hat dieser Datensatz ein **SD**-Präfix:

➔ Pluzeichen (+) neben **Score ! [Apply]** auswählen, so dass sich der **SAS_DATA_SETS** öffnet ➔ Datensatz mit **SD**-Präfix auwählen ➔ **Role** und **Description** identifiziert den Datensatz nun als **score data** ➔ **OK** ➔ Im **Data**-Tab im **Insight based on**-Feld **Entire data set** auswählen.

➔ Speichern und schließen.

➔ Rechter Mausklick ➔ Run ➔ View Results

Nun wird das Ergebnis des Score-Vorgangs präsentiert. Die Variablen **P_DINEBIN1,** d.h. der Kunde wird als Käufer identifiziert, und **P_DINEBIN0,** Klassifikation als Nicht-Käufer, geben die die Wahrscheinlichkeitswerte an. Die Werte beider Variablen ergänzen sich zu 1. Abschließend können weitere Analysen vorgenommen werden:

➔ **P_DINEBIN1** auswählen ➔ **Analyze** ➔ **Distribution**.

Übersicht über die im Beispiel verwendeten Variablen:

Name	Measurement	Model Role	Variable Label
ACCTNUM	Nominal	ID	Account Number
AGE	Interval	Input	Age
AMOUNT	Interval	Input	Dollars Spent
APPAREL	Interval	Input	Apparel Purch.
APRTMNT	Binary	Input	Rents Apartment
BLANKETS	Interval	Input	Blankets Purch.
COATS	Interval	Input	Coats Purch.
COUNTY	Interval	Input	County Code
CUSTDATE	Interval	Rejected	Date 1st Order
DINING	Interval	Rejected	Total Dining (kitch+dish+flat)
DISHES	Interval	Rejected	Dishes Purch.
DOMESTIC	Interval	Input	Domestic Prod.
EDLEVEL	Nominal	Input	Education Level
FLATWARE	Interval	Rejected	Flatware Purch.
FREQUENT	Interval	Input	Order Frequency
HEAT	Nominal	Input	Heating Type
HHAPPAR	Interval	Input	His/Her Apparel
HOMEACC	Interval	Input	Home Furniture
HOMEVAL	Interval	Input	Home Value
INCOME	Interval	Input	Yearly Income
JEWELRY	Interval	Input	Jewelry Purch.
JOB	Interval	Input	Job Category
KITCHEN	Interval	Rejected	Kitchen Prod.
LAMPS	Interval	Input	Lamps Purch.
LEISURE	Interval	Input	Leisure Prod.
LINENS	Interval	Input	Linens Purch.
LUXURY	Binary	Input	Luxury Items
MARITAL	Binary	Input	Married (y/n)
MENSWARE	Interval	Input	Mens Apparel
MOBILE	Binary	Input	Occupied <1 yr
NTITLE	Nominal	Input	Name Prefix
NUMCARS	Ordinal	Input	Number of Cars
NUMKIDS	Interval	Input	Number of Kids
OUTDOOR	Interval	Input	Outdoor Prod.
PROMO13	Interval	Input	Promo: 8-13 mon
PROMO7	Interval	Input	Promo: 1-7 mon.

Name	Measurement	Model Role	Variable Label
PURCHASE	Binary	Input	Purchase (y/n)
RACE	Nominal	Input	Race
RECENCY	Interval	Input	Recency
RETURN	Interval	Input	Total Returns
SEX	Binary	Input	Sex
STATECOD	Nominal	Input	State Code
TELIND	Binary	Input	Telemarket Ind.
TMKTORD	Ordinal	Input	Telemarket Ord.
TOWELS	Interval	Input	Towels Purch.
TRAVTIME	Interval	Input	Travel Time
VALRATIO	Interval	Input	$ Value per Mailing
WAPPAR	Interval	Input	Ladies Apparel
WCOAT	Interval	Input	Ladies Coats

A.24 Prozessbeschreibungen der Fallstudien - Fallstudie B

Eine Variablen-Übersicht mit Angaben zur Skalierung und Modellrolle sowie eine Beschreibung wird nach der Prozessbeschreibung präsentiert.

24.1 Auswahl der Netzwerkarchitektur bei NRBF-Netzen

1. Neues Projekt anlegen:

➔ **File** ➔ **New** ➔ **Project**

Projektname: *KNN*; Diagrammname: *NRBF*.

Folgendes SAS® Enterprise Miner™-Diagramm ist nötig zur Bearbeitung der Fallstudie:

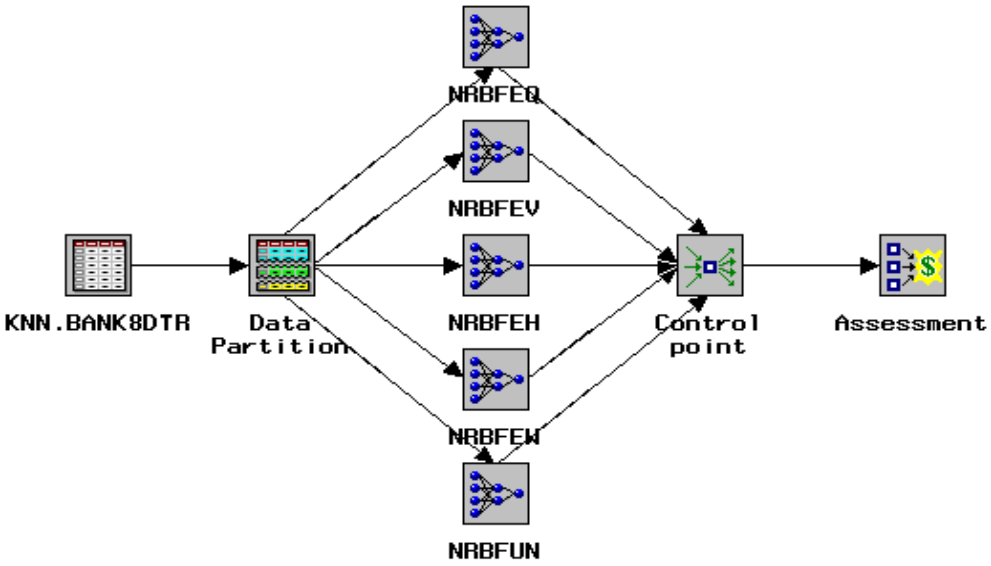


Abb. A.22.1: SAS® Enterprise Miner™-Diagramm für Fallstudie B: NRBF-Netzwerkarchitektur.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source: Target-Variable bestimmen und Prior-Vektor definieren

➔ Rechter Mausklick ➔ **Open** ➔ **Select** ➔ **KNN**-Library[7] und **BANK8DTR**-Datensatz auswählen.

Zielvariable identifizieren:

➔ Im **Variables**-Tab die Variable **ACQUIRE** auswählen ➔ Rechter Mausklick auf Spalte **Model Role** ➔ **target** auswählen

Prior-Vektor definieren:

➔ Rechter Mausklick auf **ACQUIRE**-Variable ➔ **Add** ➔ Für **Level 1** = *0.12* und für **Level 0** = *0.88* eintragen ➔ Rechter Mausklick auf **Prior vector**

➔ **Set to use** auswählen ➔ Speichern ➔ Speichern und schließen.

[7] Eine Library mit dem Namen KNN wurde erstellt. Die verwendeten Daten stammen aus dem *Neural Network Modeling Course.*

3. Data Partition: Einteilung in Trainings-, Validierungs- und Testdaten

Die Default-Einstellungen für die Trainings-, Validierungs- und Testdaten (40 / 30 / 40) werden übernommen.

4. Neural Network: Auswahl der verschiedenen Netzwerkarchitekturen

Generell:

➔ Rechter Mausklick ➔ **Open** ➔ Im **General**-Tab die Option **Advanced User Interface** auswählen.

Spezifikation der verschiedenen Netzwerkarchitekturen:

➔ Im **Advanced**-Tab im **Network**-Untertab den Schalter **Create Network** anklicken

➔ Bei **Network architecture** entsprechend der gewünschten Netzwerkarchitektur wählen: **Eq. Heights, Eq. Volumes, Eq. Widths, Eq. Widths and Heights** oder **Uneq. Widths and Heights ➔ OK.**

Einstellung der Optionen in den Schichten des KNN:

➔ Verdeckte Schicht (blaues Quadrat) anklicken ➔ Rechter Mausklick ➔ **Properties** ➔ In den **Hidden**-Tab wechseln ➔ Bei **Activation function** von **Softmax** auf **Exponential** wechseln.

➔ Output-Schicht (ACQUIRE unter gelben Symbol) anklicken ➔ Rechter Mausklick **Properties** ➔ In den **Target**-Tab wechseln ➔ Bei **Error function** die **Poisson**-Verteilung auswählen ➔ Bei **Activation function** von **Softmax** auf **Exponential** wechseln ➔ Bei **Combination function** (vgl. oben **Network architecture**) die entsprechende Netzwerkarchitektur auswählen:

Eq. Heights, Eq. Volumes, Eq. Widths, Eq. Widths and Heights oder **Uneq. Widths and Heights.**

Der letzte Schritt wird in der Hidden-Layer automatisch vorgenommen.

➔Speichern und schließen.

Model Name: *NRBFEH*; **Model Description:** *NRBF Eq. Heights.*

Model Name: *NRBFEV*; **Model Description:** *NRBF Eq. Volumes.*

Model Name: *NRBFEW*; **Model Description:** *NRBF Eq. Widths.*

Model Name: *NRBFEQ*; **Model Description:** *NRBF Eq. Heights and Widths.*

Model Name: *NRBFUN*; **Model Description:** *NRBF Unconstrained.*

5. Control Point: Vereinfachung der Strukturen

Zur Strukturvereinfachung werden die verschiedenen Netzwerke mit dem Control Point verbunden, von dem dann nur eine Verbindung zu dem Assessment-Knoten geht.

6. Assessment: Modellbewertung

Zuerst wird die Data Mining-Analyse gestartet. Alle Knoten (von der Input Data Source bis zum Assessment) werden bearbeitet.

➔ **Run** ➔ **View Results: Yes**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang.

➔ **NRBFEV-, NRBFEH-, NRBFUN, NRBFEQ-** und **NRBFEV-**Model auswählen.

➔ **Tools** ➔ **Lift Chart.**

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

24.2 Auswahl des Lernverfahren bei MLP-Netzwerkarchitekturen

1. Neues Diagramm anlegen:

➔ **File** ➔ **New** ➔ **Diagram**

Diagrammname: *NRBF*.

Folgendes SAS® Enterprise Miner™-Diagramm ist zur Bearbeitung der Fallstudie erforderlich:

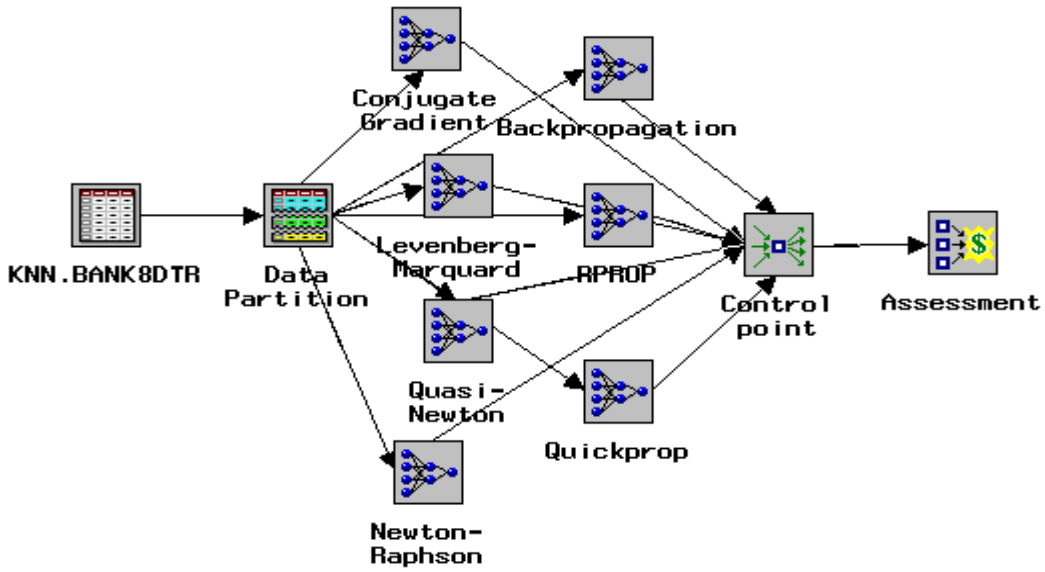


Abb. A.22.2: SAS® Enterprise Miner™-Diagramm für Fallstudie B: MLP und Lernverfahren.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source und Data Partition:

Die Einstellungen, die bei der Auswahl der Netzwerkarchitektur bei NRBF-Netzen getroffen wurden, können übernommen werden.

3. Neural Network: Auswahl der verschiedenen Lernverfahren

Generell:

➔ Rechter Mausklick ➔ **Open** ➔ Im **General**-Tab die Option **Advanced User Interface** auswählen ➔ Bei **Model selection criteria** den **Average Error** auswählen.

Spezifikation der verschiedenen Netzwerkarchitekturen:

➔ Im **Advanced**-Tab im **Network**-Untertab den Schalter **Create Network** anklicken ➔ Bei **Hidden neurons** die Anzahl der Neuronen auf *2* setzen. ➔ Bei **Direct connections Yes** auswählen ➔ OK.

➔ Auf dem Schaubild rechter Mausklick ➔ **Add hidden layer** ➔ **Connect all inputs** ➔ **Connect all targets**.

➔ Im **Optimization**-Untertab **Objective Function** auf **Maximum Likelihood** stellen.

➔ Im **Train**-Untertab die Option **Default Settings** ausstellen ➔ Nun können die verschiedenen Lernverfahren, für das jeweilige Künstliche Neuronale Netzwerk ausgewählt werden:

Conjugate Gradient, Levenberg-Marquard, Newton-Raphson, Quasi-Newton, Standard Backprop, RPROP oder **Quickprop.**

➔ Speichern und Schließen.

Model Name: *Quickpro*; **Model Description:** *MLP with Quickprop.*

Model Name: *RPROP*; **Model Description:** *MLP with RPROP.*

Model Name: *Backprop*; **Model Description:** *MLP with Backpropagation.*

Model Name: *Con.Grad*; **Model Description:** *MLP with Conjugate Gradient.*

Model Name: *Lev-Marq*; **Model Description:** *MLP with Levenberg-Marquard.*

Model Name: *New-Raph*; **Model Description:** *MLP with Newton-Raphson.*

Model Name: *Quasi-Ne*; **Model Description:** *MLP with Quasi-Newton.*

4. Control Point: Vereinfachung der Strukturen

Zur Strukturvereinfachung werden die verschiedenen Netzwerke mit dem Control Point verbunden, von dem dann nur eine Verbindung zu dem Assessment-Knoten geht.

5. Assessment: Modellbewertung

Zuerst wird die Data Mining-Analyse gestartet. Alle Knoten (von der Input Data Source bis zum Assessment) werden bearbeitet.

➔ **Run** ➔ **View Results: Yes**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang.

➔ **Quickpro-, RPROP-, Backprop-, Lev-Marq-, Con.Grad-, New-Raph-,** und **Quasi-Ne-**Model auswählen ➔ **Tools** ➔ **Lift Chart**

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

24.3 Early Stopping

1. Neues Diagramm anlegen:

➔ **File** ➔ **New** ➔ **Diagram**

Diagrammname: *NRBF.*

Folgendes SAS® Enterprise Miner™-Diagramm ist nötig zur Bearbeitung der Fallstudie:

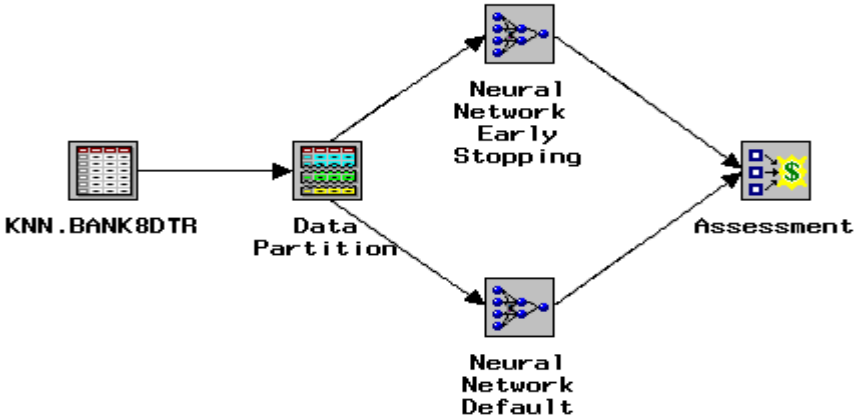


Abb. A.22.3: SAS® Enterprise Miner™-Diagramm für Fallstudie B: Early Stopping.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source: Target-Variable bestimmen

➔ Rechter Mausklick ➔ **Open** ➔ **Select** ➔ **KNN**-Library und **BANK8DTR**-Datensatz auswählen.

➔ Im **Variables**-Tab die Variable **ACQUIRE** auswählen ➔ Rechter Mausklick auf Spalte **Model Role** ➔ **target** auswählen

➔ Speichern und schließen.

3. Data Partition: Einteilung in Trainings-, Validierungs- und Testdaten

➔ Rechter Mausklick ➔ **Open** ➔ Im Feld **Train** = *90%*, im Feld **Validation** = *10%* und im Feld **Test** = *0%* eintragen ➔ Speichern und schließen

4. Neural Network: Early Stopping-Modell vs. Default-Modell

Early Stopping Modell:

➔ Rechter Mausklick ➔ **Open** ➔ Im **General**-Tab die Option **Advanced User Interface** auswählen.

➔ Im **Advanced**-Tab im **Network**-Untertab den Schalter **Create Network** anklicken ➔ Bei **Hidden neurons** die Anzahl der Neuronen auf *9* setzen.

➔ Im **Train**-Untertab die Option **Default Settings** ausstellen ➔ **Conjugate gradient** auswählen.

➔ Speichern und schließen.

Model Name: *Early Stopping*; **Model Description:** *KNN with 9 Neurons.*

Beide Modelle werden nun gestartet:

➔ Rechter Mausklick ➔ **Run** ➔ **View Results: Yes**

Im Plot-Tab sind nun die Zu- bzw. Abnahme des Validierungs- und Trainingsfehler in Abhängigkeit von der Anzahl an Iterationsschritten zu sehen.

5. Assessment: Modellbewertung

➔ Rechter Mausklick ➔ **Results**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang.

➔ **Early Stopping-** und **Dafault-**Model auswählen ➔ **Tools** ➔ **Lift Chart**

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

Übersicht über die im Beispiel verwendeten Variablen:

Name	Measurement	Model Role	Variable Label
ACQUIRE	Binary	ID	Acquire Invest. Prod.
ADBDDA	Interval	Input	Avg. Daily Balance in Check. Acc.
ATMCT	Interval	Input	Number of Transations per Month
ATRES	Interval	Rejected	Date 1st Order
DDADEP	Interval	Input	Tot. Amount of Check. Deposits
DDATOT	Interval	Input	Tot. Amount of Check. Trans.
INCOME	Interval	Input	Monthly Income
INVEST	Interval	Input	Amount of Investments
SAVBAL	Interval	Input	Saving Account Balance

A.25 Prozessbeschreibungen der Fallstudien Fallstudie C:

Eine Variablen-Übersicht, die angibt welche Skalierung, Modellrolle und Beschreibung die Variablen in dem verwendeten Beispiel besitzen, wird nach der Prozessbeschreibung präsentiert.

25.1 Bestimmung des Attributauswahlmaßes

1. Neues Projekt anlegen:

➔ **File** ➔ **New** ➔ **Project**

Projektname: *Tree*; Diagrammname: *SplitCrit.*

Folgendes SAS® Enterprise Miner™-Diagramm ist zur Bearbeitung der Fallstudie erforderlich:

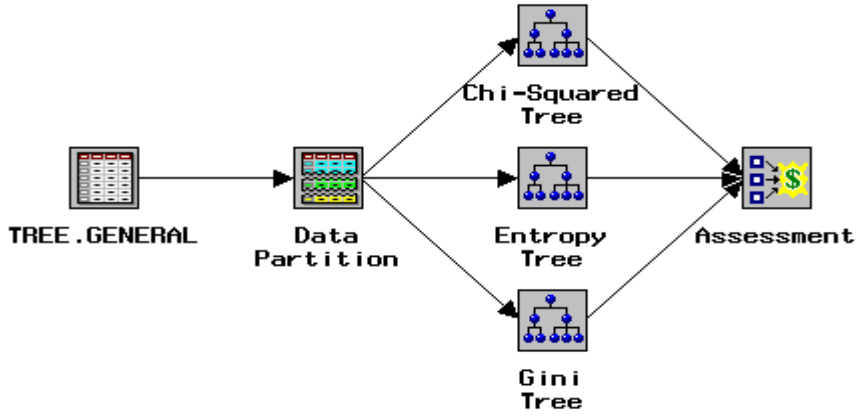


Abb. A.23.1: SAS® Enterprise Miner™-Diagramm für Fallstudie C: Attributauswahlmaß.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source: Target-Variable bestimmen und Metadaten anpassen

➔ Rechter Mausklick ➔ **Open** ➔ **Select** ➔ **TREE**-Library[8] und **GENERAL**-Datensatz auswählen.

➔ Im **Variables**-Tab die Variable **MAJORDER** auswählen ➔ Rechter Mausklick auf Spalte **Model Role** ➔ **target** auswählen.

➔ Im **Variables**-Tab die Variable **COOKIE** auswählen ➔ Rechter Mausklick auf Spalte **Model Role** ➔ **rejected** auswählen

➔ Im **Variables**-Tab die Variablen **NETYRS**, **WWWHOME**, **WWWWORK**, **WWWSCH**, **WWWPUB**, **BUDGET**, **CHILDREN**, **INCOME** und **AGE** auswählen.

➔ Rechter Mausklick auf Spalte **Measurement** ➔ **ordinal** auswählen.

[8] Eine Library mit dem Namen TREE wurde erstellt. Die verwendeten Daten stammen aus dem *Decision Tree Modeling Course.*

➔ Im **Variables-**Tab die Variablen **LANGUAGE**, **VOTE**, **INDUSTRY**, **OCCUPAT**, **SECTOR**, **MAILTO**, **EDU**, **RACE**, **MARITAL**, **LOCATION**, **AREA** und **PLATFORM** auswählen.

➔ Rechter Mausklick auf Spalte **Measurement ➔ nominal** auswählen.

➔ Speichern und schließen.

3. Data Partition: Einteilung in Trainings-, Validierungs- und Testdaten

➔ Rechter Mausklick ➔ **Open** ➔ Im Feld **Train** = *67%*, im Feld **Validation** = *33%* und im Feld **Test** = *0%* eintragen ➔ Speichern und schließen.

4. Tree: Bestimmung der verschiedenen Attributauswahlverfahren

➔ Rechter Mausklick ➔ **Open** ➔ Im **Basic**-Tab lassen sich unter der Option **Splitting criterion** die gewünschten Attributauswahlverfahren auswählen, um verschiedene Bäume mit unterschiedlichem Baumwachstum zu generieren ➔ **Chi-square test**, **Entropy reduction** oder **Gini reduction**.

➔ Im **Advanced**-Tab unter **Model assessment measure** die **Proportion of event in top 10%** auswählen.

➔ Speichern und Schließen.

Model Name: *ChiSquar*; **Model Description:** *Chi square-Tree.*

Model Name: *Entropy*; **Model Description:** *Entropy-Tree.*

Model Name: *Gini* **Model Description:** *Gini-Tree.*

Alle Modelle werden nun gestartet:

➔ Rechter Mausklick ➔ **Run ➔ View Results: Yes**

Im Entscheidungsbaumknoten werden als Ergebnisse statistische Zusammen-fassungen, ein Ring-Plot und Trainings- und Validierungswerte für die Bäume mit 1 bis N Blättern, samt Graphen präsentiert. Der Entscheidungsbaum lässt sich aufrufen, indem in der Menüleiste unter **View** der Menüpunkt **Tree** ausgewählt wird.

5. Assessment: Modellbewertung

➔ Rechter Mausklick ➔ **Results**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang.

➔ **ChiSquar**-, **Entropy**- und **Gini-**Model auswählen ➔ **Tools ➔ Lift Chart**

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

25.2 Bagging

1. Neues Diagramm anlegen:

➔ **File** ➔ **New** ➔ **Diagram**

Diagrammname: *Bagging.*

Folgendes SAS® Enterprise Miner™-Diagramm ist nötig zur Bearbeitung der Fallstudie:

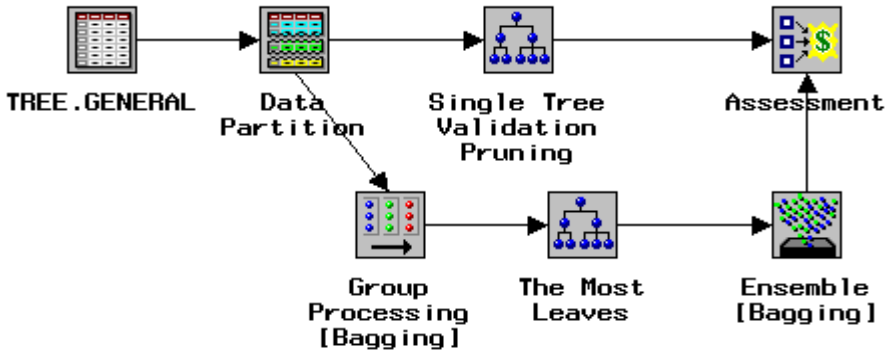


Abb. A.23.2: SAS® Enterprise Miner™-Diagramm für Fallstudie C: Bagging.
Quelle: Screenshot SAS® Enterprise Miner™.

2. Input Data Source:

Die Einstellungen, die bei der Bestimmung des Attributauswahlmaßes getroffen wurden, können übernommen werden.

3. Data Partition: Einteilung in Trainings-, Validierungs- und Testdaten

Die Default-Einstellungen für die Trainings-, Validierungs- und Testdaten (40 / 30 / 40) werden übernommen.

4. Tree: Single Tree-Einstellungen

Als Einzel-Modell wird ein Entscheidungsbaum mit Default-Einstellungen verwendet.

➔ Rechter Mausklick ➔ Modell auswählen ➔ **Options** ➔ **Training**, **Validation** und **Test** auswählen.

5. Group Processing: Bagging-Einstellungen

➔ Rechter Mausklick ➔ **Open**.

➔ Im **General**-Tab unter **Mode** die Option **Unweighted resampling for bagging** auswählen. Die **Number of loops** auf *25* erhöhen.

➔ Im **Unweighted resampling for bagging**-Tab die **Sample Size** auf *50 %* setzen.

➔ Speichern und schließen.

6. Tree: Tree für Bagging-Prozess vorbereiten

➔ Rechter Mausklick ➔ **Open**.

➔ Im **Advanced**-Tab unter **Sub-tree** die Option **The most leaves** auswählen.

➔ Speichern und schließen.

Model Name: *Bagging*; **Model Description:** *The Most Leaves.*

7. Ensemble: Bagging-Einstellungen

➔ Rechter Mausklick ➔ **Open**.

➔ Im **Settings**-Tab unter **Ensemble mode** das Verfahren **Bagging** auswählen.

Model Name: *Bagging*; **Model Description:** *Ensemble: Bagging & Tree.*

8. Assessment: Modellbewertung

Zuerst wird die Data Mining-Analyse gestartet. Alle Knoten (von der Input Data Source bis zum Assessment) werden bearbeitet.

➔ **Run** ➔ **View Results: Yes**

Außer den statistischen Auswertungen sind vor allem die Draw Lift Charts von Belang.

➔ **Tree-** und **Ensemble-**Model auswählen ➔ **Tools** ➔ **Lift Chart**.

Nun lassen sich die verschiedenen Charts für die Modellbewertung nutzen:

%Response, %Captured Response, Lift Value, Profit und **ROI.**

Übersicht über die im Beispiel verwendeten Variablen:

Name	Measurement	Model Role	Variable Label
AGE	Ordinal	Input	<5, 5-10,… 81-85, >85.
AREA	Nominal	Input	Urban, Suburban, Rural.
BUDGET	Ordinal	Input	Organization's Tot. Budget in M$ <1, 1-10,…, 500-1000, >1000.
CHILDREN	Ordinal	Input	Number of Children in Household 0=1, 1=2, 2=3, 3=4, >4=5.
COOKIE	Binary	Rejected	Changed Cookie Reference.
DISCOG	Binary	Input	Cognitive Disability.
DISHEAR	Binary	Input	Hearing Disability.
DISMOTO	Binary	Input	Motor Disability.
DISVIS	Binary	Input	Vision Disability.
EDU	Nominal	Input	Education Attainment: Grammar, High School, College, Tech, Doc. Prof., Other.
GENDER	Binary	Input	Female, Male.
INCOME	Ordinal	Input	Household Income (<$10=2, 10-19$=3, …, >100$=9).
INDURSTRY	Nominal	Input	32 Different Professional Groups.
LANGUAGE	Nominal	Input	18 Different Languages.
LOCATION	Nominal	Input	13 Different Regions.
MAILTO	Nominal	Input	Prof. Correspondence is with… Public, Private, Non-Profit, Other.

Name	Measurement	Model Role	Variable Label
MAJORDER	Binary	Target	Made a Purchase Online > $100.
MARITAL	Nominal	Input	Marital Status.
NETYRS	Ordinal	Input	Years on Internet (<0,5; 0,5-1; 1-3, 4-6, >7).
OOCUPAT	Nominal	Input	Professional Position.
PAYDAD	Binary	Input	Parents Pay for Access.
PAYSCH	Binary	Input	School Pays for Access.
PAYSELF	Binary	Input	Self Pays for Access.
PAYWORK	Binary	Input	Work Pays for Access.
PLATFORM	Nominal	Input	Primary Computing Platform.
RACE	Nominal	Input	White, Afr. Amer., Indigen., Asian, Hisp., Latino, Multi., Other.
REVAPPRO	Binary	Input	Rev. from Govt. Appropriations.
REVCGOV	Binary	Input	Rev. from Contracts with Govt.
REVCPRI	Binary	Input	Rev. from Contracts with Private.
REVDONAT	Binary	Input	Revenues from Donations.
REVSCOTH	Binary	Input	Rev. from Contracts with Other.
REVSGOV	Binary	Input	Rev. from Sales with Govt.
REVSPRI	Binary	Input	Rev. from Sales with Private.
REVUSER	Binary	Input	Revenues from User fees.
SECTOR	Nominal	Input	Public, Private, Non-Profit, Other=5.
VOTE	Nominal	Input	Registered to Vote (No=2, Yes=3, NA=4).
WWWHOME	Ordinal	Input	Access www from Home: Daily, Weekly, Monthly, >Once/Month, Never
WWWPUB	Ordinal	Input	Access www from Public: Daily, Weekly, Monthly, >Once/Month, Never
WWWSCH	Ordinal	Input	Access www from School: Daily, Weekly, Monthly, >Once/Month, Never
WWWWORK	Ordinal	Input	Access www from Work: Daily, Weekly, Monthly, >Once/Month, Never

A B B I L D U N G S V E R Z E I C H N I S

Abb. 2.1: Top-Down- und Bottom-Up-Analyse – OLAP und Data Mining 12

Abb. 4.1: Null-Modell vs. Interpolation 21
Abb. 4.2: Modellanpassung eines Entscheidungsbaumes 21
Abb. 4.3: Modellanpassung mit der Regression 21
Abb. 4.4: Einteilung von Clusteranalyseverfahren 27
Abb. 4.5: Pruning eines Entscheidungsbaumes 32
Abb. 4.6: Ensemble-Modell als Kombination von Mehrfach-Modellen 34
Abb. 4.7: Multiple lineare Regression und logistische Regression als Netzwerk-Diagramme 36
Abb. 4.8: Netzwerk-Diagramms des Feedforward Netzes bzw. des Multilayer Perzeptrons 37
Abb. 4.9: Radiale-Basisfunktionen-Netze 40
Abb. 4.10: Normalisierte Radiale-Basisfunktionen-Netze 41
Abb. 4.11: Einschränkungen bezüglich der Gestaltung des Höhen-Parameters und der Gewichte 41
Abb. 4.12: Mögliche Fehlerflächen eines Neuronalen Netzes als Funktion der Gewichte w_1 und w_2 42
Abb. 4.13: Lokales Minimum einer Fehlerfläche und Fehlerfläche mit weiten Plateaus 43
Abb. 4.14: Oszillationen in steilen Schluchten und Verlassen guter Minima 44
Abb. 4.15: Architektur einer SOM 48

Abb. 5.1: Assessment-Kriterien 52
Abb. 5.2: Confusion-Matrix 52
Abb. 5.3: %Response 53
Abb. 5.4: %Captured Response 53
Abb. 5.5: Lift Value 54

Abb. 6.1: Ausgangssituation zur Berechnung einer Profit-Matrix 56
Abb. 6.2: SAS® Enterprise Miner™-Diagramm zur Selektion der Kunden aufgrund ihrer Kaufwahrscheinlichkeiten 57
Abb. 6.3: %Response- und Profit-Chart für Regressionsanalyse und Entscheidungsbaumverfahren 57

Abb. 6.4: Beobachtungen mit der höchsten Kaufwahrscheinlichkeit 58
Abb. 6.5: NRBF-Netzwerkarchitektur 59
Abb. 6.6: Draw Lift Charts für die verschienen Normalisierten Radialen-Basisfunktionen-Netze 59
Abb. 6.7: Statistische Auswertungen für die verschiedenen NRBF-Architekturen 60
Abb. 6.8: SAS® Enterprise Miner™-Diagramm für die Bestimmung des Lernverfahrens sowie die Netzarchitektur der einzelnen KNN 61
Abb. 6.9: Bewertung der Lernverfahren: Konjugierter Gradientenabstieg, Backpropagation, Levenberg-Marquard, Quasi-Newton, Newton-Raphson, RPROP und Quickprop 61
Abb. 6.10: Modellbewertung mit Draw Lift Charts 62
Abb. 6.11: SAS® Enterprise Miner™-Diagramm für ein KNN mit einer verdeckten Schicht mit 9 Neuronen und ein Default-KNN 63
Abb. 6.12: Early Stopping-KNN vs. Default-KNN 64
Abb. 6.13: SAS® Enterprise Miner™-Diagramm für die Bestimmung des Auswahlmaßes 65
Abb. 6.14: Entscheidungsbaum mit χ^2-Auswahlmaß 65
Abb. 6.15: Training vs. Validierung von Entscheidungsbäumen mit den Auswahlmaßen χ^2, Entropy und Gini-Index 66
Abb. 6.16: SAS® Enterprise Miner™-Diagramm zur Durchführung eines Bagging-Prozesses 67
Abb. 6.17: Bagging-Modell vs. Einzelmodell 67

Abb. A.1: Ablauf eines Vorhersage- bzw. Klassifikationsmodells VII
Abb. A.2: Bias und Standardabweichung eines Funktionsschätzers f IX
Abb. A.3: Architektur von Data Warehouse, Data Mining und OLAP X
Abb. A.4.1: OLAP-Würfel mit den Dimensionen Region, Produkt und Zeit XI
Abb. A.4.2: Selektion unterschiedlicher Datensichten mittels des Slice-Verfahrens XI
Abb. A.5: Einsatzgebiete des Data Mining: Funktionale und branchenspezifische Anwendungen XII
Abb. A.6: Cross Validation XIII
Abb. A.7: Complete Case Analysis XIV

Abb. A.8: Biologisches Neuron XV
Abb. A.9.1: Logistische Funktion XVI
Abb. A.9.2: Tangens Hyperbolicus XVI
Abb. A.10: Topologien verschiedener KNN XVII
Abb. A.11: Support, Konfidenz und Lift einer Assoziationsregel (Produkt A ➔ Produkt B) XX
Abb. A.12: ROC-Kurve XXI
Abb. A.13: Die Benutzeroberfläche des SAS® Enterprise Miner™ XXIII
Abb. A.14: Die Knoten der Sample-Gruppe: Input Data Source, Sampling und Data Partition XXIV
Abb. A.15: Die Knoten der Explore-Gruppe: Distribution Explorer, Multiplot, Insight, Association, Variable Selection und Link Analysis XXV
Abb. A.16: Die Knoten der Modify-Gruppe: Data Set Attributes, Transform Variables, Filter Outliers, Replacement, Clustering, SOM / Kohonen und Time Series XXVI
Abb. A.17: Die Knoten der Model-Gruppe: Regression, Tree, Neural Network, Princomp / Dmneural, User Defined Model, Ensemble, Memory-Based Reasoning und Two Stage Model XXVII
Abb. A.18: Die Knoten der Assess-Gruppe: Assessment und Reporter XXIX
Abb. A.19: Die Knoten der Score-Gruppe: Score und C*Score XXX
Abb. A.20: Die Knoten der Utility-Gruppe: SAS Code, Control Point, Subdiagram, Group Processing und Data Mining Database XXXI
Abb. A.21: SAS® Enterprise Miner™-Diagramm für Fallstudie A XXXII
Abb. A.22.1: SAS® Enterprise Miner™-Diagramm für Fallstudie B: NRBF-Netzwerkarchitektur XXXVIII
Abb. A.22.2: SAS® Enterprise Miner™-Diagramm für Fallstudie B: MLP und Lernverfahren XL
Abb. A.22.3: SAS® Enterprise Miner™-Diagramm für Fallstudie B: Early Stopping XLII
Abb. A.23.1: SAS® Enterprise Miner™-Diagramm für Fallstudie C: Attributauswahlmaß XLIV
Abb. A.23.2: SAS® Enterprise Miner™-Diagramm für Fallstudie C: Bagging XLVI

L I T E R A T U R V E R Z E I C H N I S

Anders, U. (1995): *Neuronale Netze in der Ökonometrie*, Discussion Paper, Mannheim.

Backhaus, K.; Erichson, B.; Plinke, W. und Weiber, W. (2000): *Mutlivariate Analysemethoden: Eine anwendungsorientierte Einführung*, Berlin, Heidelberg, New York.

Badner, J. (1994): *Clusteranalyse: Eine anwendungsorientierte Einführung*, München.

Bea, F.X.; Dichtl, E. und Schweitzer, M. (1994): *Allgem. Betriebswirtschaftslehre*, Tübingen, Mannheim.

Benenati, I. (1998): *Neuronale Netze im Portfoliomanagement*, Wiesbaden.

Berry, M. and Linoff, G. (1997): *Data Mining Techniques: For Marketing, Sales and Customer Support,* New York.

Berry, M. and Linoff, G. (1997): *Mastering Data Mining: The Art and Science of Customer Relationship Management,* New York.

Bishop, C.M. (1995): *Neural Network for Pattern Recognition,* New York.

Breiman, L.; Friedman, J.H.; Olshen, R.A. and Stone, C.J. (1984): *Classification and Regression Trees*, New York.

Freitas, A. (2002): *Data Mining and Knowledge Discovery with Evolutionary Algorithms,* Berlin, Heidelberg, New York.

Friedag, H.und Schmidt, W. (2002): *Balanced Scorecard*, München.

Hastie, T.; Tibshirani, R. and Friedman, J.H. (2001): *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Berlin, Heidelberg, New York.

Hofmann, M. und Mertiens, M. (Hg.)(2000): *Customer Lifetime Value Management: Kundenwerte schaffen und erhöhen: Konzepte, Strategien, Praxisbeispiele*, Wiesbaden.

Hummel, T. und Malorny, C. (2002): *Total Quality Management,* München, Wien.

Inmon, W. (1993): *Building the Data Warehouse,* New York.

Kaplan, R.S. and Norton, D.P. (2001): *Die Strategiefokussierte Organisation,* Stuttgart.

Kohonen, T. (2001): *Self-Organizing Maps*, Berlin, Heidelberg, New York.

Krahl, D. (1998): *Data Mining*, Bonn.

Küppers, B. (1999): *Data Mining in der Praxis,* Frankfurt am Main.

Lämmel, U. und Cleve, J.(2001): *Künstliche Intelligenz: Lehr- und Übungsbuch,* Leipzig, München, Wien.

Lusti, M. (2002): *Data Warehouse und Data Mining*, Berlin, Heidelberg, New York.

Muksch, H. und Behme, W. (Hg.)(2000): *Das Data Warehouse-Konzept*: *Architektur - Datenmodelle – Anwendungen*, Wiesbaden.

Nakhaeizadeh, G. (Hg.)(1998): *Data Mining: Theoretische Aspekte und Anwendungen,* Heidelberg.

Oesterer, M. (2002); *Personalisierung im Internet*: *Optimierung der Kundenbeziehung durch Datenmanagement und Datenanalyse*, Heidelberg.

SAS Institute Inc. (2002): *Applying Data Mining Techniques Using Enterprise Miner™,* Cary.

SAS Institute Inc. (2000a): *Decision Tree Modeling,* Cary.

SAS Institute Inc. (2000b): *Getting Started with the Enterprise Miner™,* Cary.

SAS Institute Inc. (2000c): *Neural Network Modeling,* Cary.

SAS Institute Inc. (2000d): *Predictive Modeling Using Logistic Regression,* Cary.

SAS Institute Inc. (2000e): *Using Enterprise Miner™ Software: A Case Study Approach,* Cary.

SAS Institute Inc. (2001): *Warehouse Architecture,* Cary.

Schinzer, H.; Bange, C. und Mertens, H. (1999): *Data Warehouse und Data Mining: Marktführende Produkte im Vergleich*, München.

Schönleben, P. (2000): *Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsprozessen,* Berlin.

Schütte, R.; Rotthowe, T. und Holten R. (Hg.)(2001): *Data Warehouse Management-Handbuch: Konzepte, Software, Erfahrungen,* Berlin, Heidelberg, New York.

Stier, W. (1999): *Empirische Forschungsmethoden*, Berlin, Heidelberg, New York.

Winkler, P. (1997): *Empirische Wirtschaftsforschung*, Berlin, Heidelberg, New York.

Wöhe, G. (1996): *Einführung in die allgemeine Betriebswirtschaftslehre,* München.

Zell, A. (2000): *Simulation neuronaler Netze*, München, Wien.

Duden (2001): *Informatik – Ein Fachlexikon für Studium und Praxis (2001),* Mannheim, Leipzig, Wien, Zürich.

A B K Ü R Z U N G S V E R Z E I C H N I S

act	Aktivierungsfunktion
AIC	Akaike's Information Criterion
ASE	Average Squared Error
BSC	Balanced Scorecard
CART	Classification and Regression Tree
CHAID	Chi-squared Automatic Interaction Detection
CRM	Customer Relationship Management
DWH	Data Warehouse
EIS	Entscheidungsorientierte Informationssysteme
ETL	Extraction, Transformation, Loading
exp	exponential
FN	False Negative
FP	False Positive
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transport Protocol
ID3	Iterative Dichotomiser 3
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
KI	Künstliche Intelligenz
KNN	Künstliche neuronale Netze
ln	Logarithmus Naturalis
logistic	Logistische Funktion
max	Maximum
min	Minimum
MIS	Management Informationssystem
MLP	Multilayer Perceptron
MSE	Mean Squared Error
MSE_s	Systematischer Mean Squared Error
MSE_u	Unsystematischer Mean Squared Error
net	Netzeingabe-Funktion

NRBF	Normalisierte Radiale-Basisfunktionen-Netze
NRBFEH	Normalisierte Radiale-Basisfunktionen-Netze (Equal Heights)
NRBFEQ	Normalisierte Radiale-Basisfunktionen-Netze (Equal Heights and Widths)
NRBFEV	Normalisierte Radiale-Basisfunktionen-Netze (Equal Volume)
NRBFEW	Normalisierte Radiale-Basisfunktionen-Netze (Equal Widths)
NRBFUN	Normalisierte Radiale-Basisfunktionen-Netze (Unconstrained)
OLAP	Online Analytical Processing
OLS	Ordinary Least Squares
out	Ausgabefunktion
P & C-Methode	Perturb and Combine-Methode
RBF	Radiale-Basisfunktionen-Netze
RDBMS	Relationales Datenbank Management System
ROC	Receiver Operating Characteristic
ROI	Return on Investment
RPROP	Resilient Propagation
SBC	Schwarz Bayesian Criterion
SCM	Supply Chain Management
SEMMA	Sample, Explore, Modify, Model, Assess
SOM	Self-Organizing Maps
SRM	Supplier Relationship Management
tanh	Tangens Hyperbolicus
TN	True Negative
TDIDT	Top Down Induction of Decision Trees
TP	True Positive
TQM	Total Quality Management
URL	Uniform Resource Locator
Var	Varianz